

No. 15-2056

IN THE
United States Court of Appeals for the Federal Circuit

SYNOPSYS, INC.,
Appellant,
v.

MENTOR GRAPHICS CORPORATION,
Appellee.

On Appeal from the Patent and Trademark Office,
Patent Trial and Appeal Board
No. IPR2014-00287

**OPENING BRIEF AND ADDENDUM FOR
APPELLANT SYNOPSYS, INC.**

Brian P. Goldman
ORRICK, HERRINGTON &
SUTCLIFFE LLP
405 Howard St.
San Francisco, CA 94105
(415) 773-5700

Travis Jensen
Anthony Tartaglio
ORRICK, HERRINGTON &
SUTCLIFFE LLP
1000 Marsh Road
Menlo Park, CA 94025
(650) 614-7400

Eric A. Shumsky
Jeremy R. Peterman
ORRICK, HERRINGTON &
SUTCLIFFE LLP
1152 15th Street NW
Washington, DC 20005
(202) 339-8400
eshumsky@orrick.com

Counsel for Appellant

Claim 1

1. A method, comprising:
 - a) inferring the existence of a resetable memory from a behavioral or RTL level description of a semiconductor circuit; and
 - b) incorporating a resetable memory design into a design for said semiconductor circuit.

A79, col. 9:49-53 ('420 patent).

CERTIFICATE OF INTEREST

Pursuant to Federal Circuit Rule 47.4, counsel for appellant certifies the following:

1. Counsel for appellant represents Synopsys, Inc.
2. Synopsys, Inc. is the name of the real party in interest.
3. There are no parent corporations and/or any publicly held companies that own 10 percent or more of the stock of appellant Synopsys, Inc.
4. The following are the names of all law firms and the partners or associates that appeared for appellant Synopsys, Inc., before the agency and/or are expected to appear in this Court:

ORRICK, HERRINGTON & SUTCLIFFE LLP

Brian P. Goldman
Travis Jensen
Jeremy Peterman
Eric A. Shumsky
Anthony Tartaglio
William H. Wright

Date: December 23, 2015

/s/ Eric A. Shumsky

Eric A. Shumsky
Counsel for Appellant

TABLE OF CONTENTS

	Page
CERTIFICATE OF INTEREST	i
TABLE OF AUTHORITIES	iv
STATEMENT OF RELATED CASES	vii
INTRODUCTION	1
JURISDICTION	4
STATEMENT OF ISSUES.....	4
STATEMENT OF THE CASE	5
Basic Techniques For Designing A Circuit	6
Structural instantiation.....	7
Basic random logic synthesis.....	8
Inference	10
Within The Circuit, Information Is Stored In Memory	12
The Special Case Of Resettable Memory.....	15
The '420 Patent Teaches A Novel Method For Automatically Incorporating Efficient Resettable Memory Designs Into Circuits	18
Mentor's Asserted Prior Art.....	21
The Board Invalidates The Challenged Claims Based On The Combination Of Vander Zanden and Shand.....	27
SUMMARY OF THE ARGUMENT	29
STANDARD OF REVIEW.....	32
ARGUMENT	33
I. The Board's Obviousness Analysis Was Both Impermissibly Cursory, And Erroneous.	33
A. The '420 patent claims "inferring the existence of a resettable memory from a behavioral or RTL level description."	34

B.	The Board did not adequately explain how the prior art disclosed the “inferring” limitation.....	37
C.	The combination of Vander Zanden and Shand neither teaches nor suggests “inferring the existence of a resetable memory from a behavioral or RTL level description.”	46
CONCLUSION		55
ADDENDUM		
Institution of <i>Inter Partes</i> Review, dated June 12, 2014		A1
Final Written Decision, dated June 11, 2015.....		A25
U.S. Patent No. 6,836,420		A64
CERTIFICATE OF SERVICE		
CERTIFICATE OF COMPLIANCE		

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Allentown Mack Sales & Serv., Inc. v. NLRB</i> , 522 U.S. 359 (1974)	38, 43
<i>Ariosia Diagnostics v. Verinata Health, Inc.</i> , Nos. 2015-1215, -1226, 2015 U.S. App. LEXIS 19847 (Fed. Cir. Nov. 16, 2015)	40
<i>In re Bond</i> , 910 F.2d 831 (Fed. Cir. 1990)	39
<i>Burlington Truck Lines, Inc. v. United States</i> , 371 U.S. 156 (1962)	38
<i>Gechter v. Davidson</i> , 116 F.3d 1454 (Fed. Cir. 1997)	31, 38, 39, 43
<i>In re Gurley</i> , 27 F.3d 551 (Fed. Cir. 1994)	54
<i>Insite Vision, Inc. v. Sandoz, Inc.</i> , 783 F.3d 853 (Fed. Cir. 2015)	32
<i>Institut Pasteur & Universite Pierre et Marie Curie v.</i> <i>Focarino</i> , 738 F.3d 1337 (Fed. Cir. 2013)	33
<i>K/S HIMPP v. Hear-Wear Techs., LLC</i> , 751 F.3d 1362 (Fed. Cir. 2014)	32
<i>In re Kahn</i> , 441 F.3d 977 (Fed. Cir. 2006)	39, 40
<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007)	40, 42
<i>Motor Vehicle Mfrs. Ass’n v. State Farm Mut. Auto. Ins. Co.</i> , 463 U.S. 29 (1983)	38

<i>In re Mouttet</i> , 686 F.3d 1322 (Fed. Cir. 2012)	54
<i>Mullins v. Dep’t of Energy</i> , 50 F.3d 990 (Fed. Cir. 1995)	39
<i>N. Telecom Ltd. v. Samsung Elecs. Co.</i> , 215 F.3d 1281 (Fed. Cir. 2000)	45
<i>Nippon Steel Corp. v. United States</i> , 458 F.3d 1345 (Fed. Cir. 2006)	32, 33
<i>In re Nouvel</i> , 493 Fed. App’x 85 (Fed. Cir. 2012)	40
<i>In re Ochiai</i> , 71 F.3d 1565 (Fed. Cir. 1995)	33
<i>Plantronics, Inc. v. Aliph, Inc.</i> , 724 F.3d 1343 (Fed. Cir. 2013)	40
<i>Power Integrations, Inc. v. Lee</i> , 797 F.3d 1318 (Fed. Cir. 2015)	3, 31, 38, 40, 46
<i>In re Rouffet</i> , 149 F.3d 1350 (Fed. Cir. 1998)	40
<i>In re Sang-Su Lee</i> , 277 F.3d 1338 (Fed. Cir. 2002)	37, 38, 39
<i>SEC v. Chenery Corp.</i> , 318 U.S. 80 (1943)	39, 40
<i>SEC v. Chenery Corp.</i> , 332 U.S. 194 (1947)	38
<i>In re Thrift</i> , 298 F.3d 1357 (Fed. Cir. 2002)	38, 39, 41, 46
<i>Universal Camera Corp. v. NLRB</i> , 340 U.S. 474 (1951)	33

Federal Statutes & Regulations

28 U.S.C. § 1295(a)(4)(A).....	4
35 U.S.C. § 103.....	33
35 U.S.C. § 141(c).....	4
35 U.S.C. § 316(e)	33, 46
35 U.S.C. § 319.....	4
37 C.F.R. § 90.3(a)(1)	4

Other Authorities

Oxford English Dictionary Online, http://www.oed.com/view/Entry/13464	36
--	----

STATEMENT OF RELATED CASES

Pursuant to Federal Circuit Rule 47.5, appellant Synopsys, Inc. identifies *Synopsys, Inc. v. Mentor Graphics Corp.*, No. 3:12-cv-06467 (N.D. Cal.), as a related case. In that action, Synopsys has asserted U.S. Patent No. 6,836,420 (the '420 patent) against Mentor. That is the same patent whose validity Mentor has challenged in the inter partes review at issue here. The district court has stayed that action pending resolution of this inter partes review.

INTRODUCTION

Imagine that you hire an architect to help you design a new home. You describe to the architect in functional terms what you would like the house to include. You might say, for example, that you want to be able to control the temperature in each room. The architect could design any number of systems in response. She could use very simple components to achieve the specified function—for instance, a thermometer and a space heater in each room, which you would have to switch on and off as you go from room to room. Or your architect might infer that you are describing a centralized zone-heating system. After recognizing this, she could design a central furnace with air ducts into each room that are controlled by room-specific thermostats. You probably would be happier with the architect in the second scenario, because she has inferred what you were asking for and implemented an efficient design in response.

This case is about just such a process of inference, but in the context of designing circuits rather than homes. Synopsys' '420 patent teaches a method by which automated circuit-design tools (the virtual architects in this domain) can figure out from a functional (or

“behavioral”) description written by a circuit designer that a particular circuit function is needed—namely, resetting a memory—and then implement an efficient mechanism for providing this function, like the zone-heating system. The patent refers to this as “inferring the existence of a resetable memory from a behavioral or RTL level description.” A79, col. 9:49-50. This critical step helps a circuit-design tool to avoid implementing a basic design out of primitive components—akin to the space heaters in every room—that would perform the same function less elegantly or efficiently.

In the inter partes review here, the Board held the challenged claims, each of which contain this “inferring” limitation, to be obvious. It did so based on the combination of two references, Vander Zanden and Shand. Yet these references do not teach or suggest “inferring the existence of a resetable memory from a behavioral ... description,” either alone or in combination, and the Board failed to explain how they do. Instead, the Board said little more than that each of the claim elements can be found in these references—that Vander Zanden discloses inferring whether something is a memory, and Shand discloses an example of a resetable memory. But even if that were true, it

doesn't establish that it is obvious to infer resetable memory from a behavioral description. The Board offered little more than a bare conclusion, which is not a "reasoned explanation" sufficient to enable meaningful judicial review. Accordingly, the decision must be vacated and remanded. *Power Integrations, Inc. v. Lee*, 797 F.3d 1318, 1323 (Fed. Cir. 2015).

Had the Board seriously engaged the claim language, it could not have found "inferring the existence of a resetable memory from a behavioral ... description" to be obvious. The elements are not all present in the prior art. Vander Zanden's method does not disclose resetable memory, and it certainly does not disclose inferring the existence of a resetable memory. Instead, it describes a method that allows circuit designers to *expressly* direct a synthesis tool to implement a basic memory, and such express direction is the opposite of inferring. But even if Vander Zanden did know how to "infer" the *basic* memories that are the subject of that article, inferring *resetable* memory—a concept Vander Zanden does not discuss—is an altogether different matter. The fact that Shand discloses a circuit structure for a resetable memory tells us nothing about whether or how a person of ordinary

skill in the art could combine Vander Zanden, with Shand, to *infer* the need for such resetable memory from a behavioral description. In fact, one could not. So even if the Board's bare conclusion were sufficiently reasoned even to be judicially reviewable, the decision should be reversed.

JURISDICTION

Mentor petitioned for inter partes review of claims 1-3, 10-13, and 20 of the '420 patent. On June 11, 2015, the Board issued a final written decision cancelling all challenged claims. A25-63. On August 12, 2015, Synopsys timely filed its notice of appeal. A598-604; 35 U.S.C. §§ 141(c), 319; 37 C.F.R. § 90.3(a)(1). This Court has jurisdiction under 28 U.S.C. § 1295(a)(4)(A).

STATEMENT OF ISSUES

1. Whether the Board's decision fails to enable judicial review because it does not adequately explain how the combination of Vander Zanden and Shand teaches or suggests "inferring the existence of a resetable memory from a behavioral or RTL level description."

2. Whether the Board erred in holding the challenged claims to be obvious given the absence of evidence that the combination of Vander

Zanden and Shand teaches or suggests “inferring the existence of a resetable memory from a behavioral or RTL level description.”

STATEMENT OF THE CASE

Designing circuits for a modern-day computer chip is an expensive, time-consuming, and high-stakes process. The circuits that run everything from microwave ovens to cars must constantly be made smaller, and use less power, as they take on more and more sophisticated tasks.

This poses a challenge. For years, chip designers would have individually designed each and every circuit. That no longer is possible, given the size and complexity of many modern chips. Instead, chip designers rely on specialized, computer-aided design software known as “synthesis tools,” which permit the designer simply to describe the *operations* the circuit should perform. These descriptions are known as “behavioral” descriptions. The tool then translates this behavioral description into the blueprint for a circuit design. Like all forms of artificial intelligence, though, synthesis tools are imperfect, and they will not always implement an optimal design. The ’420 patent

innovated a solution to this problem for one specialized memory component called a “resetable memory.”

We begin by describing the process of designing circuits, then turn to the '420 patent and the prior art.

Basic Techniques For Designing A Circuit

When an engineer designs a circuit, she does not typically draw up a schematic. Instead, she writes a description of how the desired circuit should behave. The engineer writes this “behavioral circuit description[]” in a Hardware Description Language (HDL), such as VHDL or Verilog. A2702 ¶ 27 (Declaration of Dr. Hutchings, Synopsys’ expert). These languages are often referred to generically as Register Transfer Level (RTL) languages. *Id.* Thus, “[a] behavioral level or RTL level description is a circuit description that is tailored to be understood by a computer and that describes the circuit in terms of its methodology ... as opposed to describing the circuit only in terms of its hardware components and the interconnections between them.” A77, col. 5:37-44 ('420 patent). In our heating example, a behavioral description might

be code describing heating in response to the temperature falling below the thermostat setting.

This behavioral description then must be translated into a schematic for the circuit (akin to a blueprint in our house analogy). That translation process is called “synthesis.” Because a behavioral description “does not specify the actual implementation or the methodology used to implement” the design, the synthesis tool needs to figure out which components will achieve the desired function. A2707

¶ 35. Like an architect who translates the client’s desired functions (“ability to control temperature”; “sunny, spacious room”; “wheelchair accessible”) into a set of plans, a synthesis tool reads the behavioral description and translates it into the wires and electrical components needed to implement the desired function in an actual circuit. See A2702 ¶ 27. The tool can do this in any of three general ways: “structural instantiation,” “basic random logic synthesis,” or “inference.”

Structural instantiation. This first method is a straightforward one by which a circuit designer instructs the synthesis tool, in express terms, which circuit design to use. Thus, when a designer “structurally instantiat[es]” a circuit, she not only describes

the circuit's behavior, but also specifies in the design code the circuit's components and structure. A653 (Vander Zanden); *see also* A2716-17 ¶ 47 (Hutchings declaration). In other words, the synthesis tool need perform no translation at all because the designer has told it, in express terms, what to do. The synthesis tool then incorporates this precise design into the blueprint. In our heating example, the homeowner herself would tell the architect, "I would like a heating system that contains a multi-zone furnace with air ducts to every room in the house that are controlled by room-specific thermostats."

This approach eliminates much of the efficiency of using a synthesis tool in the first place. As Vander Zanden puts it, when structural instantiation is used, "the designer must manually complete the design and cannot take full advantage of the synthesis tools." *See* A653.

Basic random logic synthesis. A second method for translating a behavioral description into a circuit is to provide the synthesis tool with a general set of rules that are designed to perform translation on any description. *See* A653 (Vander Zanden); A2715-16 ¶ 46 (Hutchings declaration). This is called "basic random logic synthesis." *See* A653.

Using this method, a synthesis tool follows the rules, and implements the behavioral design in a piecemeal fashion by putting together basic circuit components that will perform each individual function described in the RTL source code. In short, a synthesis tool employing this translation technique tends to look at each individual aspect of the source code description.

This is necessary, as the rules are designed to work for a large number of functions, but it also leads to a significant drawback. A synthesis tool employing this technique will not recognize, for instance, that multiple lines of a behavioral description, when taken as a whole, correspond collectively to a circuit component that is not built efficiently in a piecemeal fashion, but instead could be better implemented by a single, “specific block of larger circuitry.” A2715-16 ¶ 46; *see also* A653 (Vander Zanden) (basic random logic synthesis “cannot take advantage of special architectures”).

In short, the tool simply implements each piece of the behavioral description separately, without understanding whether and how each piece fits into the bigger picture. An architect employing this elementary thinking would implement our heating system with

something like a space heater and thermometer in each room. That would satisfy the literal requirement that it be possible to make the temperature constant in each room, but it wouldn't do so practically or efficiently. Nor would the homeowner likely be happy with the result.

Inference. This final approach to translating a behavioral description into a structural representation of a circuit (such as a schematic) can be thought of as a more thoughtful and useful form of basic random logic synthesis. The synthesis tool will be programmed to “analyze[] the behavior” described by the programming code, and “determine[] that the behavior ... , as a whole, corresponds to a specific block of larger circuitry.” A2703, ¶ 28 (Hutchings declaration); *see also* A2715-17 ¶¶ 46-48 (same). In other words, the tool figures out that the code's functional description corresponds to some category of hardware component. This makes it possible for the tool to incorporate a special, more efficient design for the component into the circuit. A2716-17 ¶ 47. Where inference is concerned, the key is for the synthesis tool to determine—to “infer”—that a specific behavior can and should be implemented as a coherent block of circuitry. A2703, 2715-16 ¶¶ 28, 46 (“as described and claimed by the '420 patent, the synthesis tool could

be written so that it is capable of inferring that [a] block of RTL code, as a whole, specifically describes a memory with reset”).

The equivalent is the architect’s realization that the homeowner’s functional description describes a zone-heating system and that the system could be satisfied by a central furnace with air ducts to each room controlled by room-specific thermostats. The design that ultimately is implemented will be the same as in our structural instantiation example, but the “inference” approach is far more useful because the synthesis tool figures out and creates the structural implementation without having to be told directly by the homeowner. A77, col. 6:23-28 (’420 patent); A653 (Vander Zanden); A2716-17 ¶ 47 (Hutchings declaration).

All of these methods may be used by a synthesis tool (albeit with varying results) to create a schematic showing the circuit structure. The invention in this case concerns one of these methods in particular (inference), as applied to translating a particular circuit feature: a resetable memory. We begin by discussing memory generally, then turn to the special case of resetable memory.

Within The Circuit, Information Is Stored In Memory

Often a designer will write a behavioral description that calls for a circuit to remember information. A circuit can do this in a number of ways. One of the most basic is a storage component called a “flip-flop.” A flip-flop is a simple circuit element that can store a single bit of information, usually represented by a 1 or a 0. A2698 ¶ 17 (Hutchings declaration). Another storage component, similar to a flip-flop, is a “latch.” It differs, for purposes of this case, only in that it takes up less surface area. *See* A2584 (Detjens deposition).

More complex circuits require the ability to store many bits of information simultaneously. One way to accomplish this is by arranging multiple flip-flops into a “register.” A2698-99 ¶ 18 (Hutchings declaration). A register typically holds a closely related group of bits (for instance, a single numerical value) that is too large to store in a single flip-flop. *Id.*

Registers, in turn, can be grouped together to form a “memory” (although using groups of registers is an inefficient way to construct a memory). *See* A2699 ¶ 19 (Hutchings declaration). A memory, more generically, is “a device for storing data having a plurality of cells, each

cell having a unique address for storing data, where data is written to a cell during a write function, and, during a read function data is retrieved from a cell and presented at a memory output.” A35 (Board’s claim construction); *see also* A2698-700 ¶¶ 19-21 (Hutchings declaration). And because a smaller chip is generally preferable to a larger one, designers typically prefer to use specialized “memory cores” (which are pre-designed compact memory units), rather than memories assembled from basic components like flip-flops and registers, which are bigger and slower. *See* A2700-01 ¶¶ 22-23 (Hutchings declaration).

This diagram depicts a typical semiconductor memory core:

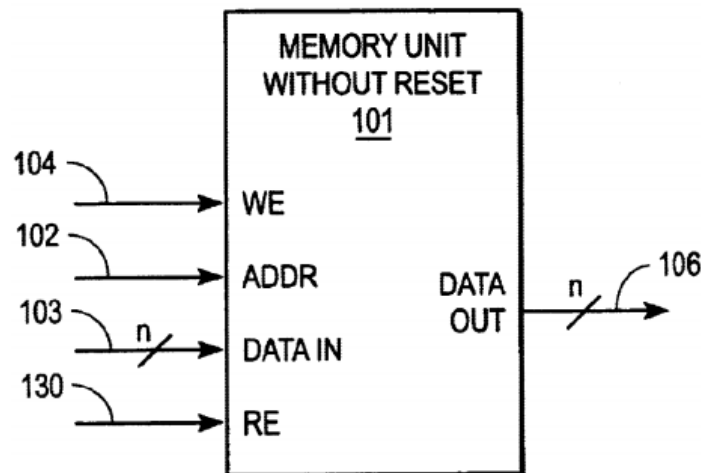


FIG. 1

A65 ('420 patent) (Fig. 1).

In a traditional memory core like this one, data is written to and read from the memory one cell at a time. For a user to write to a cell within the memory, the memory needs to know several things (though the order of operations may vary). *First*, it needs to know that it is being asked to store new information. This step is triggered by sending a “write enable” (WE) signal to the chip, via the WE line (104). *Second*, the memory needs to know where that information should be stored. This is achieved by telling the memory via the ADDR line (102) the address (ADDR) of the cell to be written to. *Third*, the memory needs to know what information should be written to the memory. This is accomplished by transmitting, via the Data In line (103), the data that are to be written to the specified address. A75, col. 1:21-29 (’420 patent). To rewrite new information into a cell, the same process is followed.

Then, when it comes time to *retrieve* stored data, the memory must be told to read data from a particular location. A “read enable” (RE) signal is supplied to the RE line (130) to alert the memory that it should perform the “read” operation. A75, col. 1:32-37. The memory is also told, along the memory’s ADDR line (102), the address of the

relevant cell from which to read information. *Id.* And finally, the “word” of stored data is delivered along the Data Out line (106). A75, col. 1:37-38.

The Special Case Of Resettable Memory

Sometimes a functional design calls for resetting all of the cells in a memory to a “reset value” (e.g., 0). This is called a “reset,” and it can be used to ensure that the memory does not output information that has become outdated. This function may be performed with a “resettable memory,” and the ’420 patent describes a number of different circuit designs that can be used to create a resettable memory. Some are slow and take up a lot of space. Others are highly efficient in the right circumstances and therefore more desirable.

In one primitive design, a resettable memory could be created in piecemeal fashion by assembling a large bank of flip-flops, with each flip-flop storing one bit of data and having the capability to be “individually accessed via complicated” circuitry and individually reset. Just as each room might have its own heater in our architectural example above, here, each bit of data has its own reset function. A75, col. 1:60-67. And just as a space heater in every room is a piece-by-piece

implementation that fails to appreciate the relationship between the individual components and the larger structure, this primitive design is too. This approach, the '420 patent explains, “can result in a resetable memory unit having noticeably slower performance (and that consumes more silicon surface area).” A75, col. 2:2-5.

Alternatively, a designer could use a memory with a reset capability built into an efficient memory core. Such a memory has an additional “reset” input line, and can effectively and quickly set all of the memory’s cells to a reset value. A75, col. 1:59-63. This is called a “resetable memory” with “intrinsic reset.” A2701-02 ¶ 24 (Hutchings declaration). But these units were not widely available. *See* A2707-08 ¶ 36.

The patent describes two better designs. When the performance of a reset operation is not time-constrained, a circuit can be configured with a “reset value write unit,” a piece of circuitry that will write a reset value (e.g., 0) to each and every address within the memory, one by one, going through the steps outlined above (at 14): Enable the ability to write data, specify the address as location 1, write the value “0”; repeat the same process for address 2; and so on. A72 ('420 patent) (Fig. 6);

A78, col. 8:10-16; *see also* A2714-15 ¶ 45 (Hutchings declaration). This approach is time-consuming, although it has the advantage of taking up very little surface area on the chip. *See* A79, col. 9:16-21.

When time is limited, and a reset must occur quickly, all of the cells must be reset at once. The '420 patent describes a method for doing so that achieves the benefits of a memory core, while diminishing expense, by assigning the reset function to a piece of the circuit located outside of the memory. Specifically, data storage is accomplished by using a highly efficient memory core *without* reset, rather than the slow and bulky array of flip-flops described above. That memory core is paired with a small, separate memory unit that has intrinsic reset (or equivalent circuitry). In combination, these devices “emulate” a fully resettable memory through a clever trick. The small unit with reset acts as a sort of index, to keep track of which addresses have been reset but not yet written over. And it communicates this information to the rest of the circuit by outputting a signal—typically a zero—any time one of those addresses in the memory core has been reset (but not yet rewritten with new data), even though the actual memory core continues to hold old data at that address. A76, col. 3:3-23, 50-60 ('420

patent); A2708-10 ¶¶ 37-39 (Hutchings declaration). Through this clever workaround, a circuit can store information in the efficient memory core, while the addition of a very small, intrinsically resetable memory gives the illusion that the entire memory is resetable.

The '420 Patent Teaches A Novel Method For Automatically Incorporating Efficient Resetable Memory Designs Into Circuits.

A behavioral description that calls for resetting a memory can be implemented in any of the ways described above, but some implementations will be better than others. And when it comes to picking the right implementation, a design tool will be most useful to a designer if, like our architect, it can automatically figure out the best implementation. *See* A2716-17 ¶ 47 (Hutchings declaration) (“Automatic inference greatly eases the effort required to use the described resettable-memory invention.” (citing A77, col. 5:9-13)). But, before assessing *which* resetable memory design to use, the tool first must be able to determine that resetable memory is needed—i.e., that the behavioral description corresponds to a larger block of circuitry that can be implemented using one of the efficient resetable memory designs. Think back to our example of zone heating. Before determining the *best* zone-heating system for a house, the architect first

must figure out that, when the homeowner says she wants to control the temperature in every room, this functional description best corresponds to zone-heating systems, rather than a primitive design like a space heater in every room.

This is where the '420 patent comes in. It teaches a method for the tool to “infer,” that is, figure out, that the behavioral description corresponds to a resetable memory. This is the crucial first step for the tool to incorporate one of the efficient designs into the circuit. For in order to select among the various possible resetable memory designs, the tool must first recognize that the behavioral description requires the circuit component known as resetable memory. Only then can the tool choose between and automatically incorporate different designs for that resetable memory component. The patented method accordingly makes it “eas[y]” for designers to incorporate an efficient resetable memory design into a circuit design, saving them valuable time and effort. A77, col. 5:10-15. Claim 1 is representative:

1. A method, comprising:

- a) inferring the existence of a resetable memory from a behavioral or RTL level description of a semiconductor circuit; and

b) incorporating a resetable memory design into a design for said semiconductor circuit.

A79, col 9:47-53. With regard to the inferring step, the specification explains that

automatic inference can be accomplished ... by configuring the design tool to recognize from the operational flow of the circuit that: 1) some type of reset is being applied to the stored data values within the circuit; and 2) the stored data values are being changed to some type of reset value in response.

A77, col. 6:22-27.

In other words, the invention scans the behavioral description for a particular function that the designer wants the circuit to perform (namely, wiping out stored data by means of a reset), and figures out that what the designer has described in functional terms corresponds to a resetable memory. Then, having thereby inferred that the behavioral description corresponds to a resetable memory, the design tool is able to “modif[y]” the blueprint that it would ordinarily implement using basic random logic synthesis, and thereby it “incorporate[s] the resetable memory.” A77, cols. 5:60-6:6. And in claims not at issue here, the patent builds upon this “inferring” and further teaches how the

synthesis tool can select the most efficient implementation. A79, col. 10:12-28 (claims 8 and 9); *see also* A73 (Fig. 7).

But without the “inferring” step, that would not be possible. If a synthesis tool lacked the ability to figure out that a behavior described in the RTL code corresponded to a resetable memory, it could only blindly implement the piecemeal, primitive design that “basic random logic synthesis” would generate. And in that case, the synthesis tool would simply implement the bank of individually resetable flip-flops or latches discussed above (at 15-16) that the ’420 patent criticized as the inefficient prior art approach. *See* A75, cols. 1:60-2:6; A584-85 (Mentor’s statement from oral hearing); A2715-16 ¶ 46 (Hutchings declaration). Without first figuring out that a behavioral description corresponds to a resetable memory, the tool would have no basis to automatically implement one of the efficient resetable memory designs.

Mentor’s Asserted Prior Art

Mentor petitioned for inter partes review. A167-232. Relevant here, the Board instituted review to consider whether claims 1-2, 10-12, and 20 were anticipated by Vander Zanden, and whether claims 1-3,

10-13, and 20 were obvious based on the combination of Vander Zanden and Shand.¹

Vander Zanden. Vander Zanden is a short paper that teaches a multi-step process “for synthesizing small memories” that enables designers to minimize the size of the storage unit. A653.

Recall that circuits can store information using basic components such as flip-flops, latches, or registers, and that designers typically prefer more efficient storage components. *Supra* 12-13. Vander Zanden recognized that existing synthesis methods made it difficult for designers to use the more efficient components. It explains that, when presented with a behavioral description aimed at storing information, a synthesis tool ordinarily uses “basic random logic synthesis” (*supra* 8-10)—and that this approach will simply “generate a design consisting of flip-flops/latches,” which are inefficient storage units. A653. Vander Zanden further explains that “structural instantiation”—the approach by which the designer tells the design tool which hardware design to

¹ The Board also instituted review on certain other references, but ultimately found those references not to be invalidating. A26.

implement, *supra* 7-8—is undesirable because “the designer must [then] manually complete the design.” *Id.*

Vander Zanden proposed a hybrid approach: Let the designer expressly direct the synthesis tool to build a memory with a specific, more efficient, memory storage unit, but leave the remainder of the design to whatever the tool generates using basic random logic synthesis. A654. The designer’s directions to the tool come in the form of a “synthesis comment,” *id.*—e.g., “use a register.” The tool then performs synthesis, and incorporates the designated style of storage unit.

Thus, the tool described in Vander Zanden is indifferent to how components of the memory are implemented, other than the storage component specified by the designer. It leaves those other characteristics to random logic synthesis, without figuring out whether the memory as a whole corresponds to a larger component, such as a resetable memory, as would be necessary for it to automatically implement a more efficient design. A654-55. Because the method relies on both user directives and synthesis from a behavioral description, it

amounts to a hybrid between random logic synthesis and structural instantiation. *Id.*

Vander Zanden also recognizes its own limitations. If its method “fail[s] to meet the designer’s needs,” it instructs designers to use “structural instantiation of a pre-designed module”—the very method it criticized as undesirable because the designer does the thinking for the tool. A653; *cf.* A77, col. 5:9-19 (’420 patent) (discussing “further utility” of the ’420 patent’s different method).

Mentor asserted that Vander Zanden is anticipatory based on one line of code in a figure, and the accompanying circuit diagram. A107. Specifically, Vander Zanden’s Figure 3 includes an example of a behavioral description, and that description happens to include a reset:

```

architecture regfilesyn_A of regfilesyn is
begin
  process(clk,rst)
    subtype addrType is integer range 0 to 15;
    type regFileType is array (addrType) of integer;
    variable regFile: regFileType;
  begin
    if rst='1' then
      out1 <= (others => '0');
    elsif clk'event and clk='1' then
      if s2='1' then
        out1 <= regFile(addr1) + regFile(addr2);
      else
        out1 <= regFile(addr3);
      end if;
      if s1='1' then
        regFile(writeAddr) := dataIn;
      end if;
    end if;
  end process;
end regfilesyn_A;

```

Figure 3: VHDL Description

A655 (coloring added). According to Mentor, this behavioral description, contained in a snippet of RTL code, describes resetting a memory, and the resulting circuit thus contains a resetable memory (specifically, on the output flip-flop DFF6, which is the second component highlighted in yellow):

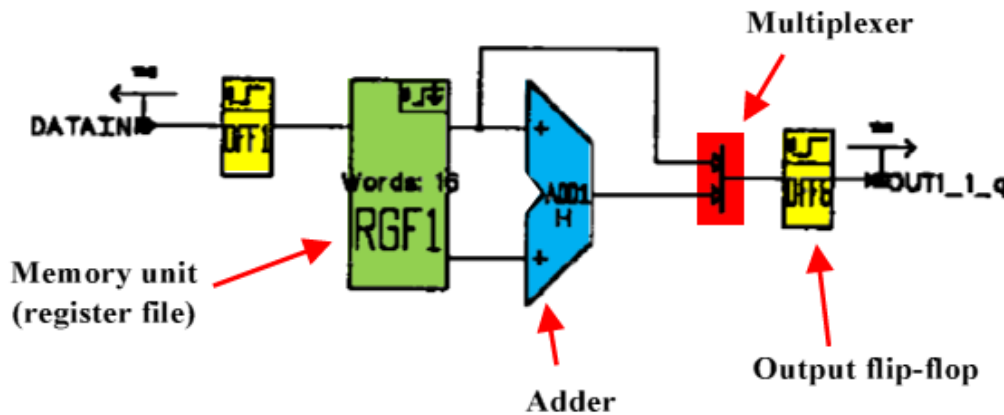


Figure 4: Datapath Design Containing Register File Cell

A2726 ¶ 65 (Hutchings mark-up of Vander Zanden Figure 4). See A189 (Mentor’s petition for inter partes review).

Dr. Hutchings demonstrated, however (and the Board ultimately agreed, A45), that the source code in Vander Zanden does not describe a resettable memory. Hutchings explained that, for a memory to be resettable, it must continue to output the reset value until new data is written to the memory. A2724-25 ¶ 63. But the “rst” signal in Vander Zanden Figure 3 does not do so. A45. It puts out a “0” (indicating a reset) only so long as the rst signal is continuously and repeatedly sent to the circuit. It therefore is not a resettable memory, because it does not always continue to output a reset signal until new data has been written to the storage unit.

Shand. Mentor further relied on Shand as disclosing a specific embodiment of a resetable memory. A125-29. Shand is a short patent that discloses a method and apparatus for resetting a memory (which, as it happens, is the same one discussed above, at 17-18). *See* A1007, col. 1:35-60. But that is all Shand does. Shand does not discuss synthesis. It does not discuss the process of synthesizing memories. It does not discuss using synthesis to incorporate its memory design into a circuit. And it does not discuss *inferring* a resetable memory during the process of synthesizing a memory from a behavioral description. A2637 (Detjens deposition) (“Q: Does the Shand patent discuss anything about inferring memories? A. No.”); A2736 ¶ 86 (“Shand doesn’t discuss inference of the existence of memory with reset from RTL. Shand essentially consists of a few circuit diagrams with explanations of their behavior.”). Shand’s disclosure is limited to its particular resetable memory design. A1007, col. 1:35-60.

The Board Invalidates The Challenged Claims Based On The Combination Of Vander Zanden And Shand.

Ultimately, the Board found the challenged claims obvious. The Board first agreed with Synopsys’ expert, Dr. Hutchings, that a “resetable memory” must “output[] a reset value until new data is

written into the memory.” A39. That interpretation of “resetable memory” doomed Mentor’s anticipation argument, because the reset function Mentor pointed to in Vander Zanden does not continue outputting a reset value until new data is written. The Board therefore held that Vander Zanden fails to anticipate because it “does not disclose a ‘resetable memory.’” A45.

But in the course of analyzing anticipation, the Board found that Vander Zanden *does* teach “‘inferring’ ... a memory with reset capability” because “in one state downstream circuitry would recognize the circuit of Figure 4 as a memory.” A44. The Board did not explain whether or how Vander Zanden draws this inference “from a behavioral or RTL level description,” as the claims require. *See* A79, col. 9:49-50; *compare* A44 (final written decision).

This passing statement about “inferring” in the context of anticipation provided the basis for the Board’s conclusion of obviousness. The Board simply combined its determination that Vander Zanden disclosed “inferring a memory” with Shand’s disclosure of a “resetable memory,” and declared that the combination rendered the challenged claims obvious. A48, 50 (emphasis omitted). According

to the Board: “As discussed previously ... with respect to the anticipation challenge to Vander Zanden, we determined that Vander Zanden discloses inferring a ‘memory’ in accordance with our claim construction, but not a ‘resetable memory.’” A48. And it stated that it would have been “within the ordinary skill of one in the art to have modified the respective underlying HDL code ... to include code defining the respective resetable memory circuit such as disclosed in Shand.” A50. How doing so would teach or perform the step of *inferring* the need for *resetable* memory, in response to this newly included code, the Board did not say.

SUMMARY OF THE ARGUMENT

The Board erred in holding the challenged claims obvious. The prior art does not teach or suggest “inferring the existence of a resetable memory from a behavioral or RTL level description,” as each challenged claim requires.

A. “Inferring the existence of a resetable memory from a behavioral or RTL level description” is an essential step in the ’420 patent’s method for efficiently synthesizing resetable memories. Synthesis is the process of translating a behavioral description of a

design into a circuit schematic showing the components that implement the design. And to achieve the step of inferring, a synthesis tool must figure out, based on what is conveyed indirectly by the circuit's behavioral description, that the physical circuit should include a resetable memory. The inferring step is critical, for it makes it possible for the synthesis tool to implement an efficient design for a resetable memory—rather than the inefficient design based on flip-flops or latches that it would otherwise generate.

B. The Board did not meaningfully or adequately explain how it found the “inferring” limitation met. Instead, it merely said that each element of the challenged limitation—inferring, resetable memory, and a behavioral description—appears in the prior art, and then asserted that it would have been obvious to combine the references to achieve the claim. That form of cursory analysis and conclusory decision making is insufficient as a matter of law. It is a fundamental principle of administrative law—which applies to the Board no less than to every other administrative agency—that agencies must engage in reasoned decision making. And as part of doing so, the Board must “fully and particularly set out” the bases for its decision in order to enable

meaningful judicial review, among other reasons. *See Power Integrations*, 797 F.3d at 1323; *Gechter v. Davidson*, 116 F.3d 1454, 1457 (Fed. Cir. 1997). And when it fails to do so, as it did here, its decision must be vacated. *Id.*

C. If the Court does reach the merits, it should reverse, because Vander Zanden and Shand are not obviously combined to collectively disclose “inferring the existence of a resetable memory from a behavioral ... description.” Mentor pointed to Vander Zanden as disclosing the “inferring” limitation. But the Board found the “inferring” limitation met only by having the wrong actor do the inferring—the resulting circuitry, not the synthesis tool—and by reading the behavioral description out of the patent altogether. In fact, Vander Zanden does not teach or suggest inferring memories from a behavioral description at all, let alone resetable memories; it describes a method by which a designer can expressly direct a synthesis tool to incorporate a specific memory storage unit into a design.

Moreover, and most fundamentally, Vander Zanden does not disclose inferring a *resetable* memory. Inferring requires figuring out that a behavioral description corresponds to a particular block of

circuitry. Nothing in Vander Zanden provides a basis for inferring resetable memory. And the fact that Shand discloses a resetable memory—i.e., that it shows one particular design for such a memory—doesn’t mean that combining Shand with Vander Zanden would disclose how to *infer* a resetable memory from a behavioral description written in the RTL code. The Board did not explain how Vander Zanden’s ability to infer a generic memory means that it could infer the special form of resetable memory; it simply asserted that this was obvious. Accordingly, the decision of the Board should in the alternative be reversed.

STANDARD OF REVIEW

This Court reviews the Board’s “legal conclusions *de novo*, and [its] factual findings underlying those determinations for substantial evidence.” *K/S HIMPP v. Hear-Wear Techs., LLC*, 751 F.3d 1362, 1364 (Fed. Cir. 2014) (internal citations omitted). “Obviousness is a question of law, based on underlying factual determinations....” *Insite Vision, Inc. v. Sandoz, Inc.*, 783 F.3d 853, 858 (Fed. Cir. 2015). “Substantial evidence is more than a scintilla....” *Nippon Steel Corp. v. United States*, 458 F.3d 1345, 1351 (Fed. Cir. 2006) (alteration and quotation

marks omitted). When reviewing for substantial evidence, the “court must consider the record as a whole, including that which ‘fairly detracts from its weight,’ to determine whether there exists ‘such relevant evidence as a reasonable mind might accept as adequate to support a conclusion.’” *Id.* (quoting *Universal Camera Corp. v. NLRB*, 340 U.S. 474, 477-78 (1951)); *Institut Pasteur & Universite Pierre et Marie Curie v. Focarino*, 738 F.3d 1337 (Fed. Cir. 2013) (vacating PTO decision for lack of substantial evidence).

ARGUMENT

I. The Board’s Obviousness Analysis Was Both Impermissibly Cursory, And Erroneous.

Mentor was the petitioner, and as such, it bore the burden of proving unpatentability. 35 U.S.C. § 316(e). To prove obviousness, it was required to show that “the differences between” the challenged claims of the ’420 patent “and the prior art are such that the claimed invention as a whole would have been obvious before the effective filing date of the claimed invention to a person having ordinary skill in the art.” 35 U.S.C. § 103. This requires “a searching comparison of the claimed invention—including all its limitations—with the teaching of the prior art.” *In re Ochiai*, 71 F.3d 1565, 1572 (Fed. Cir. 1995). Here,

however, the Board did not sufficiently explain its reasoning to enable judicial review. The heart of its ruling is a single, bare-bones sentence that contains no meaningful explanation. And, indeed, the Board's unsupported assertion is wrong, as the record plainly reflects.

To explain why, we first summarize the limitation that is at issue in each of the challenged claims. § A. We then show that the Board failed to issue a sufficiently reasoned decision, as principles of administrative law required it to do. § B. And finally, we explain that, even had the Board undertaken a legally sufficient analysis, substantial evidence did not show that the combination of Vander Zanden and Shand disclosed “inferring the existence of a resetable memory from a behavioral or RTL level description.” § C.

A. The '420 patent claims “inferring the existence of a resetable memory from a behavioral or RTL level description.”

Each challenged claim contains the following step as part of the patented method:

inferring the existence
of a resetable memory
from a behavioral or RTL level description.

E.g., A79, col. 9:45-50 (claim 1) (line breaks added). This step, the patent makes clear, is part of circuit synthesis—the process of translating a functional (or “behavioral”) description of a circuit into a schematic that shows an actual, physical circuit that will achieve the desired functions. A77, col. 5:33-44; *supra* 6-7 (discussing synthesis).

This claim limitation has three basic parts, demarcated by the line breaks added above. Taken in reverse order, they are as follows. A “behavioral or RTL level description” is programming code describing the function a designer wants the circuit to perform. *Supra* 6-7.

(Within the claim limitation, this is the thing from which something must be inferred.) Next, a “resetable memory” is a type of memory that can be instructed to output a reset value for any storage cell until new data are written into that cell. *Supra* 15-18. (This is the thing whose existence must be inferred from the behavioral or RTL level description.) And finally, “inferring,” as the Board construed it, means “‘to form an opinion or reach a conclusion through reasoning and information,’ or ‘to convey an idea *indirectly*.’” A9 (claim construction in institution decision) (emphasis added) (citation omitted); A33 (final written decision; adopting the earlier construction).

So, putting the pieces back together: “inferring a resetable memory from a behavioral or RTL level description” means that the synthesis tool must figure out, based on what the circuit’s behavioral description conveys indirectly, that the circuit is meant to include a resetable memory.

The patent is clear that it is the synthesis tool that does the work; the designer merely writes the behavioral code. This is so because the synthesis tool draws inferences from the behavioral description, which does not spell out what form the physical circuit should take, rather than from a human directive that does. *See* A79, col. 9:49-50; A586 (explanation by Mentor at the oral hearing that a behavioral description won’t spell out circuit design); A2707 ¶ 35 (Hutchings declaration) (same). Thus, the specification repeatedly and consistently refers to “inferring” as a process that is “automatic,” a word well understood to mean without additional assistance. *See* A77, col. 5:33-37; *see also* A77 cols. 5:60-63, 6:28-31; A2715-17 ¶¶ 46-47 (Hutchings declaration) (“The inclusion of automatic inference is an important part of the ’420 patent.”); *cf.* Oxford English Dictionary Online, <http://www.oed.com/view/Entry/13464> (defining automatic as “self-

acting” and “with little or no direct human control”). Thus, the synthesis tool infers from a behavioral description by determining that a “reset is being applied to the stored data values,” and that “the stored data values are being changed to some type of reset value in response,” A77, col. 6:24-27—a process that occurs without human direction.

This process of drawing an inference from a behavioral description without involving the designer is critical because it is how the patented method saves the designer valuable time and effort while still producing an efficient circuit design. *See* A77, col. 5:9-11. Without the “inferring” step, the designer is left with lesser alternatives like basic random logic synthesis and structural instantiation, which, as discussed above (at 7-10), either mechanically and blindly create a simple, often inefficient design, or require the designer to take the additional time to describe the hardware design.

We turn now to the Board’s treatment of the “inferring” limitation.

B. The Board did not adequately explain how the prior art disclosed the “inferring” limitation.

1. “Tribunals of the PTO are governed by the Administrative Procedure Act, and their rulings receive the same judicial deference as do tribunals of other administrative agencies.” *In re Sang-Su Lee*, 277

F.3d 1338, 1342 (Fed. Cir. 2002). But this deference is not a blank check; it must be earned by complying with essential requirements of administrative law. *Id.* at 1345. Not only must the Board engage in “reasoned decisionmaking.” *Allentown Mack Sales & Serv., Inc. v. NLRB*, 522 U.S. 359, 374 (1974). It also must “fully and particularly set out the bases upon which it reached [its] decision,” *Power Integrations*, 797 F.3d at 1323, in order “to enable [this] court, without resort to speculation, to understand [its] reasoning,” *Gechter v. Davidson*, 116 F.3d at 1457. Thus, like all agencies, the Board must “articulate a satisfactory explanation for its action including a ‘rational connection between the facts found and the choice made.’” *Motor Vehicle Mfrs. Ass’n v. State Farm Mut. Auto. Ins. Co.*, 463 U.S. 29, 43 (1983) (quoting *Burlington Truck Lines, Inc. v. United States*, 371 U.S. 156, 168 (1962)).

This “simple but fundamental rule of administrative law,” *SEC v. Chenery Corp.*, 332 U.S. 194, 196 (1947) (“*Chenery II*”), is no mere ministerial requirement. It “promotes sound results.” *Allentown Mack*, 522 U.S. at 375. It is necessary to “allow accountability.” *In re Thrift*, 298 F.3d 1357, 1364 (Fed. Cir. 2002). And it is essential to enable “effective appellate review.” *Power Integrations*, 797 F.3d at 1323

(citing *SEC v. Chenery Corp.*, 318 U.S. 80, 94 (1943) (“*Chenery I*”).

Articulating the reasons for a decision enables the reviewing court “to determine whether [the Board] applied the law correctly and whether the evidence supported the underlying and ultimate fact findings.”

Gechter, 116 F.3d at 1457.²

This requirement is especially important where obviousness is concerned. After all,

[m]ost inventions arise from a combination of old elements and each element may often be found in the prior art. However, mere identification in the prior art of each element is insufficient to defeat the patentability of the combined subject matter as a whole. Rather, to establish a *prima facie* case of obviousness based on a combination of elements disclosed in the prior art, the Board must articulate the basis on which it concludes that it would have been obvious to make the claimed invention.

In re Kahn, 441 F.3d 977, 986 (Fed. Cir. 2006) (internal citations omitted). Thus, to guard against the ever-present risk of hindsight bias, “[r]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal

² See also *In re Sang-Su Lee*, 277 F.3d at 1342; *In re Thrift*, 298 F.3d at 1364; *Mullins v. Dep’t of Energy*, 50 F.3d 990, 992 (Fed. Cir. 1995); *In re Bond*, 910 F.2d 831, 833 (Fed. Cir. 1990).

conclusion of obviousness.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007) (quoting *In re Kahn*, 441 F.3d at 988).

Simply put, “the board’s patentability analyses must be both ‘clearly disclosed and adequately sustained.’” *Power Integrations*, 797 F.3d at 1323 (quoting *Chenery I*, 318 U.S. at 94). If “the necessary reasoning is absent, [this Court] cannot simply assume that ‘an ordinary artisan would be awakened to modify prior art in such a way as to lead to an obviousness rejection.’” *Plantronics, Inc. v. Aliph, Inc.*, 724 F.3d 1343, 1354 (Fed. Cir. 2013) (quoting *In re Nouvel*, 493 Fed. App’x 85 (Fed. Cir. 2012); *see also Ariosa Diagnostics v. Verinata Health, Inc.*, Nos. 2015-1215, -1226, 2015 U.S. App. LEXIS 19847, at *17 (Fed. Cir. Nov. 16, 2015) (vacating a Board decision because the Court could not “confidently discern” whether the decision rested on a “legally proper ground”). To the contrary, if the Board fails to “explain the reasons one of ordinary skill in the art would have ... combine[d] [the references] to render the claimed invention obvious,” then this Court will “infer that the Board used hindsight to conclude that the invention was obvious.” *In re Kahn*, 441 F.3d at 986 (quoting *In re Rouffet*, 149 F.3d 1350, 1359 (Fed. Cir. 1998).

That is precisely what happened here.

2. The critical claim limitation was “inferring the existence of a resetable memory from a behavioral or RTL level description.” *Supra* § A. In its petition, and throughout trial, Mentor maintained that the combination of Vander Zanden and Shand disclosed this limitation.

A207. Synopsys explained that it did not. A399-403, 553-55, 559. In the final written decision, the Board failed to “document its reasoning” for how the prior art taught or suggested this limitation. *In re Thrift*, 298 F.3d at 1364.

The Board’s analysis contains nothing more than the following logic. Vander Zanden “discloses inferring a ‘memory’ ... , but not a ‘resetable memory.’” A48. Shand discloses one physical embodiment of a resetable memory. A49-50. Thus, the prior art as a whole teaches “inferring the existence of a resetable memory.” A50. Missing entirely from the Board’s determination is any connective tissue, any reasoning that supports the conclusion or holds it together. The Board did not explain how it leaped from its premise (that Vander Zanden inferred a memory) to its conclusion that the art as a whole teaches or suggests inferring a *resetable* memory. Nor did the Board explain why it

analyzed whether Vander Zanden inferred a memory and not a *resetable memory*, when Mentor never urged such a theory of invalidity. The Board's only discussion of the Vander Zanden-Shand combination simply assumes the point. Here's what the Board said:

[I]t would have been within the abilities of a person of ordinary skill in the art to combine Shand's resetable memory circuit design with Vander Zanden and consequently modify the HDL code to address the functional changes imparted by the implementation of a different resetable memory in such a combination.

A50.

But the Board failed entirely to explain how this taught the claim. *Cf. KSR*, 550 U.S. at 418 (warning against mere "conclusory statements"). And certainly it is not self-evident that the combination itself obviously taught the limitation. Even if Vander Zanden disclosed inferring an ordinary memory from the designer's behavioral description, *but see infra* § C.1, simply pairing Vander Zanden with Shand's resetable memory does not demonstrate that Vander Zanden will *infer* a resetable memory. The fact that one of ordinary skill would have been aware of examples of resetable memory (which is part of the *resulting* circuitry, not the behavioral description) says nothing about

whether Vander Zanden's synthesis tool would be able to figure out that the behavioral description corresponded to such a memory.

Nor is the decision adequately supported by the Board's bare conclusion that a designer could modify the HDL code to describe a resetable memory. A50. This is for the simple reason that, as Mentor itself recognized, the code is the input, and it "is not going to be spelling out the Shand design." A586. So even if it were true that the code could be modified to describe resetable memory, this would tell us nothing whatsoever about whether the synthesis tool would then *infer* the existence of such memory. Simply put, knowing the code doesn't tell us whether the existence of a resetable memory will be inferred.

"[A]bsent from the Board's opinion is any explanation for whether, how, and why" Vander Zanden would infer the existence of a resetable memory from a behavioral description when paired with Shand.

Gechter, 116 F.3d at 1460.

To the extent other portions of the Final Written Decision might be read to speak to this issue, they "do not produce rationality but compound[] confusion." *Allentown Mack*, 522 U.S. at 376. Perhaps, giving the Board the benefit of every doubt, it thought that it had

resolved this issue sufficiently when it considered anticipation. *See* A59 (“To the extent [Mentor] alleges that the ‘inferring’ step was obvious in view of Vander Zanden in the Reply, we determined that Vander Zanden disclosed this step in our anticipation analysis.”). Leave aside that an anticipation analysis could not resolve this obviousness combination. Even if it could, the Board’s explanation of why Vander Zanden taught “inferring a memory” again consisted of a solitary sentence: “At least in one state *downstream circuitry would recognize* the circuit of [Vander Zanden] Figure 4 as a memory.”³ A44 (emphasis added). This conclusory assertion again is unsupported by any citation to the record, and again is insufficient to support the decision for that reason.

Moreover, the assertion makes no sense. It misunderstands both the actor doing the inferring, and the thing from which an inference is drawn. As to the latter, the patent teaches “inferring ... *from a behavioral or RTL level description.*” A79, col. 9:49-50 (emphasis added). But on the Board’s reasoning, the inference would be drawn

³ As discussed above, Vander Zanden’s Figure 4 illustrates a circuit generated using Vander Zanden’s method. *Supra* 24-26 (discussing A70 (Fig. 4)).

from “the circuit of Figure 4”—that is, from a formed circuit, not from the behavioral description. A44. That is the first mistake. And, as to the actor doing the inferring, in the ’420 patent it is the *synthesis tool* that infers, not circuitry. *E.g.*, A77, col. 5:53-57. But on the Board’s reasoning, it is “downstream circuitry” that “would recognize” a memory. A44.

These errors are fundamental. For the invention to work, an inference must be drawn from the behavioral description, because that is how the synthesis tool figures out that the designer’s design can be achieved with an efficient resetable memory. *See* A77, col. 5:53-59. But on the Board’s account, an inference would be drawn from a completed circuit design—which means that the inferring step would occur only *after* the behavioral description has been converted to a physical circuit. And if that were the case, inferring would serve no purpose. The Board’s conclusory assertion leads to the unmistakable conclusion that the Board fundamentally “misunderstood the invention at hand.” *N. Telecom Ltd. v. Samsung Elecs. Co.*, 215 F.3d 1281, 1287 (Fed. Cir. 2000).

Ultimately, the Board's failure to explain how the prior art taught or suggested inferring a resetable memory is perhaps understandable. As Judge Bisk explained at the oral hearing, Mentor's papers "seem to leave it up to [the Board] to figure out how to combine [the references]." A539. But that being so, Mentor simply failed to carry its burden. See 35 U.S.C. § 316(e). Instead, the Board now has passed the buck to this Court to attempt to divine what the Board might have concluded based on what Mentor didn't explain. That is precisely what the APA forbids. Because the Board provided this Court "with an inadequate predicate upon which to evaluate its decision," its decision must be vacated. *Power Integrations*, 797 F.3d at 1325; *In re Thrift*, 298 F.3d at 1366.

C. The combination of Vander Zanden and Shand neither teaches nor suggests "inferring the existence of a resetable memory from a behavioral or RTL level description."

It was with good reason that, as Judge Bisk recognized, Mentor left it to the Board to combine the references: The showing of obviousness simply cannot be made. Should the Court reach this question, it should reverse the decision and hold the challenged claims non-obvious.

Mentor contended that Vander Zanden and Shand together disclose the challenged claims. In Mentor's hypothetical combination, Vander Zanden's job is to infer a resetable memory from a behavioral description. *See* A207 ("Using the method described in Vander Zanden[, Shand's] ... memories would be inferred"). Shand's role is to disclose a *resetable* memory that Vander Zanden can incorporate into a circuit. Beyond that, all agreed, Shand plays no other part. *See* A207-09 (Mentor's argument that Shand teaches resetable memory designs).

Before even reaching the question of whether it would be obvious to combine Vander Zanden with Shand, the obviousness argument fails because the Board's decision does not support its conclusion that Vander Zanden infers memories. But even if Vander Zanden did infer memory, it cannot infer the existence of a *resetable* memory, and indeed it teaches away from doing so. For each of these reasons, the Board's decision is not supported by substantial evidence, and must be reversed.

1. As discussed above, the Board concluded in the course of analyzing anticipation that Vander Zanden teaches inferring. *Supra* 28-29. Specifically, the Board said that Vander Zanden "inferred" memory because "[a]t least in one state *downstream circuitry* would

recognize the circuit of [Vander Zanden] Figure 4 as a memory.” A44 (emphasis added). This was a theory that Mentor had not argued, and for good reason: It has no grounding in the patent.

In the '420 patent, it is the *synthesis tool* that does the inferring, not downstream circuitry. *E.g.*, A77, col. 5:24-28, 53-57, 60-63 (describing “software design tool” as doing the inferring); A2715 ¶ 46 (referring to “inference” as a process conducted by “a synthesis tool”). The downstream circuitry is the *result* of the synthesis process, not its driver. And that is not the only problem with the Board’s conclusion. The patent requires that the inference be drawn “*from a behavioral description,*” not from a fully formed circuit. *See* A79, col. 9:49-50 (emphasis added); *supra* 19-21, 34-37. But in the Board’s explanation, the behavioral description is nowhere to be found: The “inferring” occurs from a fully formed circuit by a component of that circuit.

Indeed, Vander Zanden simply does not teach inferring memories. To see why, it is useful to begin by considering the problem that Vander Zanden purported to solve. Designers wanted to use logic synthesis to incorporate simple memories into circuit designs, but standard techniques for “basic random-logic synthesis” (*supra* 8-10) could not

“take advantage of” more efficient designs for storing information. *See* A653. Instead, basic random logic synthesis would simply implement a generalized, one-size-fits-all approach using inefficient flip-flops. *Id.*

Vander Zanden’s solution was to treat “the memory portion of the design ... separately ... from the other random logic and datapath elements.” *Id.*; *see also* A656. This made it possible for the circuit designer to *direct* the synthesis tool to implement a memory with a particular type of memory storage unit—for example, a latch or a register—while relying on the tool to implement the remaining pieces of the circuit’s architecture using random logic synthesis techniques. *See* A653-54.

For exactly that reason, this process does not involve “inferring” anything: Vander Zanden depends on *the designer* to figure out that the behavioral description describes a memory and to direct the synthesis tool to implement the memory with a special architecture. *See id.* (explaining that when the designer does not include a “synthesis comment” along with code describing a memory function, the tool simply applies “standard logic synthesis techniques”). The designer must tell the tool which particular type of memory to implement by

adding a “synthesis comment” in the source code. *Id.* If the designer does not write a “synthesis comment ...[,] no special memory synthesis techniques are applied.” *Id.* Unlike in the ’420 patent, Vander Zanden’s synthesis tool never figures out on its own that a behavioral description corresponds to a “block of circuitry” that can and should be implemented with a special technique.

For this reason alone, Vander Zanden does not disclose “inferring,” and the combination fails to render the ’420 patent obvious.

2. Even if Vander Zanden taught “inferring ... a memory,” substantial evidence does not support the conclusion that combining Vander Zanden with Shand would teach “inferring the existence of a *resetable memory* from a behavioral or RTL level description.” Indeed, Vander Zanden directly teaches away from inferring nonstandard memories, like a resetable memory.

We begin again with Vander Zanden because, according to Mentor, that is the part of the combination that discloses “inferring.” *See* A207. As the Board recognized, “Vander Zanden does not disclose a resetable memory.” A45. Vander Zanden “says nothing about reset functionality.” A2722 ¶ 59 (Hutchings declaration). And “[r]esetable

memories, as a topic, do not appear in Vander Zanden.” *Id.* The Board’s logic appears to have been that, because Vander Zanden can infer memory, it also can infer resetable memory. *See* A50. But there is simply no reason to believe (and the Board identified none) that Vander Zanden’s invention can read a behavioral description, and “infer”—that is, discern from what is implicit in the description without explicitly being told—that the description corresponds to a resetable memory. That is fatal to obviousness, because such “inferring” is an element of the claim and a necessary prerequisite for Vander Zanden to be able to incorporate an efficient resetable memory design, like the Shand design, into the circuit. A77, col 5:33-37 (’420 patent) (describing incorporating a resetable memory *in response to* inferring); A2715 ¶ 46.

What the Board seems to have assumed—without ever holding, much less supporting or explaining—is that the ability to infer one feature at a high level of generality (here, a memory that stores data) necessarily encompasses the ability to infer a very specific species of that feature (a resetable memory). But as a logical matter, this just isn’t true. It simply begs the critical question—whether Vander Zanden’s form of inference would include the ability to infer the missing

feature. This is like saying that the ability to infer that an animal has eyes necessarily means being able to infer that an animal has *green* eyes. Of course this isn't so. You can infer that an animal has eyes from knowing, for instance, that it is a mammal. But you would need to know much more to be able to infer that a particular animal's eyes were green. You can infer that a vehicle is a car without being able to infer that it is an electric car. You can infer that a building is a factory without knowing whether it is a factory that makes candy or one that makes widgets.

And in just that same way, inferring the existence of a memory is not the same as, nor an obvious variant on, inferring the existence of a *resetable* memory. And again, the Board pointed to nothing saying otherwise. An ordinary memory entails, simply put, the storage of information in a plurality of individually accessible cells, *supra* 12-13; A35 (final written decision) (defining memory), and so to infer from the behavioral description the need for an *ordinary* memory, that may be all the synthesis tool needs to discern. But the '420 patent explains that inferring a *resetable* memory requires the synthesis tool to identify something more and different—"that some type of reset is being applied

to stored data values” and that the “values are being changed to some type of reset value in response.” A77, col. 6:24-27.

Accordingly, if a designer applied Vander Zanden’s method to a behavioral description in which the designer had specified applying a reset to a memory storage unit, nothing in Vander Zanden would figure out that the description as a whole corresponded to a resettable memory. Vander Zanden would therefore be unaware that the description could be implemented with a more efficient design. *See* A77, col. 6:2-6, 23-31 (’420 patent) (explaining that figuring out that a behavioral description corresponds to a resettable memory is a precondition to implementing a special design). And it would simply use basic random logic synthesis to implement the prior-art flip-flop approach—the very one that the ’420 patent surpasses, *see* A75, col. 2:63-67. *See* A2715-16 ¶ 46 (Hutchings declaration) (describing implementation where “the synthesis does not recognize the existence of a resettable memory”). Mentor effectively acknowledged this during the oral hearing. A585 (if Vander Zanden “had ... received the behavioral code describing a resettable memory, ... it might have implemented it using just the design that is shown in, for example, the patent at Column 1, the prior art design or resettable

flip-flops”); *see also* A587 (“Vander Zanden ... could then synthesize [a resetable memory] as a bunch of flip-flops.”).

3. Not only does Vander Zanden fail to teach or suggest inferring the existence of a resetable memory—it expressly teaches away from it. The reference “le[ads] in a direction divergent from the path that was taken by the applicant,” *In re Gurley*, 27 F.3d 551, 553 (Fed. Cir. 1994), thereby “preclud[ing] a determination that the reference renders a claim obvious,” *In re Mouttet*, 686 F.3d 1322, 1333 (Fed. Cir. 2012).

If Vander Zanden’s synthesis method “fail[s] to meet the designer’s needs”—as we just showed it would for a designer who wants the tool to incorporate a resetable memory—then Vander Zanden directs designers to use “[s]tructural instantiation.” A653. In short, in the very circumstances at issue here, Vander Zanden would use structural instantiation, but that approach is the very opposite of inferring. Structural instantiation requires the designer to “manually complete the design” by spelling out the physical description of the circuit. *Id.*; *supra* 7-8. The designer describes the specific design it wants the tool to implement. And the tool follows the designer’s express instructions. By directing designers to structurally instantiate circuits

for which Vander Zanden’s approaches would not “meet the designer’s needs,” Vander Zanden teaches away from the ’420 patent’s solution—in which “the use of a resetable memory is *automatically* inferred” from a behavioral description and the tool can “offer the insertion of a resetable memory” in response. A77, col. 5:26-28, 5:33-35 (emphasis added). Whereas Vander Zanden directs designers to use structural instantiation, the ’420 patent teaches them to configure a tool to figure out the circuit design from the “behavioral or RTL level description.” A79, col. 9:49-50. This “teaching away” is yet another reason why a person of ordinary skill in the art would not consider the ’420 patent’s invention obvious in light of the combination of Shand and Vander Zanden.

CONCLUSION

For the foregoing reasons, the Board’s decision should either be reversed outright, or, in the alternative, vacated and remanded.

December 23, 2015

Respectfully submitted,

/s/ Eric A. Shumsky

Eric A. Shumsky
Orrick, Herrington & Sutcliffe LLP
1152 15th Street NW
Washington, DC 20005
(202) 339-8400
eshumsky@orrick.com

Counsel for Appellant

TABLE OF CONTENTS

ADDENDUM

Institution of <i>Inter Partes</i> Review, dated June 12, 2014	A1-24
Final Written Decision, dated June 11, 2015.....	A25-63
U.S. Patent No. 6,836,420	A64-80

**Institution of *Inter Partes* Review,
Dated June 12, 2014 (Paper No. 9)**

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MENTOR GRAPHICS CORPORATION
Petitioner

v.

SYNOPSYS, INC.
Patent Owner

Case IPR2014-00287
Patent 6,836,420 B1

Before JENNIFER S. BISK, SCOTT A. DANIELS, and
PHILIP J. HOFFMANN, *Administrative Patent Judges*.

DANIELS, *Administrative Patent Judge*.

DECISION
Institution of *Inter Partes* Review
37 C.F.R. § 42.108

Case IPR2014-00287

Patent 6,836,420 B1

I. INTRODUCTION

A. *Background*

Mentor Graphics Corporation (“Petitioner”) filed a corrected petition to institute an *inter partes* review of claims 1-3, 10-13, and 20 of U.S. Patent No. 6,836,420 B1 (“the ’420 patent”). Paper 6 (“Pet.”). Patent Owner, Synopsys, Inc., timely filed a preliminary response. Paper 7 (“Prelim. Resp.”). We have jurisdiction under 35 U.S.C. § 314.

The standard for instituting an *inter partes* review is set forth in 35 U.S.C. § 314(a):

THRESHOLD – The Director may not authorize an *inter partes* review to be instituted unless the Director determines that the information presented in the petition filed under section 311 and any response filed under section 313 shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.

Upon consideration of the petition and the preliminary response, we determine that the information presented in the petition establishes a reasonable likelihood that Petitioner would prevail in showing the unpatentability of all the challenged claims. Accordingly, pursuant to 35 U.S.C. § 314, we institute an *inter partes* review for claims 1-3, 10-13, and 20 of the ’420 patent.

B. *Additional Proceedings*

Petitioner indicates that the ’420 patent is presently asserted against Petitioner in *Synopsys, Inc. v. Mentor Graphics Corporation*, Case No. 3:12-cv-06467-MMC in the U.S. District Court for the Northern District of California (“the district court case”). Pet. 1; Paper 5.

Case IPR2014-00287

Patent 6,836,420 B1

C. The '420 Patent

The '420 patent (Ex. 1001) generally relates to a method and apparatus for resettable memory and a corresponding digital circuit design. The '420 patent states that the circuitry for a resettable memory unit is complicated and expensive where “each n wide storage cell is implemented with resettable flip-flops that are individually accessed via complicated multiplexing and control circuitry.” Ex. 1001, 1:65-67. According to the patent, such a resettable memory unit is relatively slow and consumes more silicone surface area than a non-resettable memory unit. *Id.* at 2:3-6.

A solution to these challenges, proposed by the '420 patent, is designing resettable memory 220 as a combination of memory unit without reset 201, and memory unit with reset 205. *Id.* at 1:61–2:2. Figure 2A of the '420 patent, illustrating memory 220, is reproduced below.

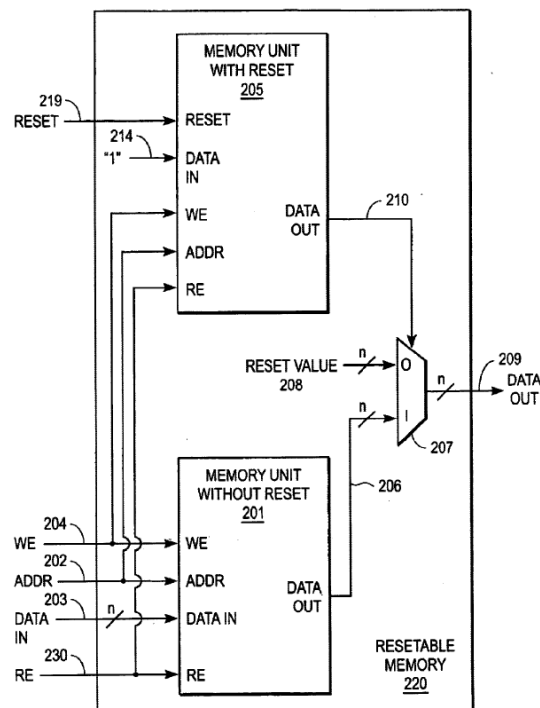


FIG. 2A

Case IPR2014-00287
Patent 6,836,420 B1

As depicted by Figure 2A of the '420 patent, above, both memory unit without reset 201 and memory unit with reset 205 are connected to data out line 209 via multiplexor 207. Memory unit without reset 201 is larger in that it has a greater data width than memory unit with reset 205. *Id.* at 3:3-10. The '420 patent describes the smaller resettable memory having a cell word size less than the cell word size of the non-resettable memory. For example, the cell word size of the resettable memory is only one bit wide, so that the memory output from a cell is either a "1" or "0." *Id.* at 3:8-10. As explained below, the purpose of the combination is that the smaller memory with reset 205 "cost effectively disguises the inability of the larger memory 201 to reset its cells." *Id.* at 3:12-13.

Resettable memory 220 operates generally as follows: data is provided through data in line 203 to memory without reset 201, and is stored in a particular cell within the memory. *See id.* Fig. 2A. The '420 patent refers to data called from memory without reset 201 as "actual memory unit data output 206." *Id.* at 3:26-30. Memory unit with reset 205 has a corresponding cell, in which a "1" from data in 214 is stored whenever WE 204 of resettable memory 220 is activated. *Id.* at 4:1-3. The "1" from the memory unit with reset 205 instructs multiplexer 207 to output the actual memory unit data output 206 from memory 201 to data out line 209. *Id.* at 4:30-35. If memory unit with reset 205 is reset, a reset value, for example "0," is stored in memory unit 205, output to data out 210, and received by multiplexer 207. In this circumstance, reset value 208 is output to data out line 209. *Id.* at 3:51-54. If no new data has been written to the particular cell in memory unit without reset 201 since its last reset, the multiplexer will

Case IPR2014-00287

Patent 6,836,420 B1

continue to output the reset value “0”¹, which remains in its storage cell until WE 204 line is activated again. *Id.* at 4:1-3. Accordingly, when a new value is written to the particular cell in memory unit without reset 201, the “0” will be over written with a “1” and the new value will be output by multiplexer 207 to data out line 209. *Id.* at 4:30-35. The ’420 patent clarifies that “[i]n this manner, the circuit of FIG. 2[A] emulates the behavior of a memory unit having the storage capacity of memory unit 201 but also having reset capability.” *Id.* at 4:46-48.

In other words, the memory unit with reset 205 replicates the behavior of a resettable memory by providing “0” or another reset value, if there has been a reset of a corresponding cell in memory unit with reset 205, unless and until new data is written to memory 201. *See id.* at 4:46-54.

In addition to the resettable memory circuit embodiment of Figure 2A discussed above, the ’420 patent describes a circuit design methodology inferring the use of a resettable memory from the behavioral level, or RTL (register-transfer level) description of the memory. *Id.* at 5:32-34. Rather than describe the specific circuit hardware, the RTL level description describes the memory circuit in terms of its function, or operational flow, including the characteristic of the memory that it is resettable. *Id.* at 5:36-44. A software design tool, by way of example, which facilitates circuit design, infers from the RTL level description that a reset condition is being applied to at least one specific variable, and the software design tool

¹ The reset may be a value besides “0,” for instance FIG. 2B discloses an embodiment in which “a reset value function circuit 230 may be inserted between (and coupled to) the reset value 208 input of the multiplexer 207 and the address input 202 of the resettable memory 220.” Ex. 1001, 4:67–5:3.

Case IPR2014-00287

Patent 6,836,420 B1

can incorporate a resettable memory into the designer's circuit design.² *Id.* at 6:27-29; *see also* Fig. 4.

D. Illustrative claims

Of the challenged claims, the independent claims are 1 and 11. Each of dependent claims 2, 3, and 10 depends directly from claim 1. Each of dependent claims 12, 13, and 20 depends directly from claim 11. Claims 1 and 11 illustrate the claimed subject matter and are reproduced below:

1. A method, comprising:
 - a) inferring the existence of a resettable memory from a behavioral or RTL level description of a semiconductor circuit; and
 - b) incorporating a resettable memory design into a design for said semiconductor circuit.

11. A machine readable medium having stored thereon a sequence of instructions which, when executed by a digital processing system, cause said system to perform a method, said method, comprising:
 - a) inferring the existence of a resettable memory from a behavioral or RTL level description of a semiconductor circuit; and
 - b) incorporating a resettable memory design into a design for said semiconductor circuit.

² The '420 patent states that "[t]he automatic inference can be accomplished, for example, by configuring the design tool to recognize from the operational flow of the circuit that: 1) some type of reset is being applied to the stored data values within the circuit; and 2) the stored data values are being changed to some type of reset value in response." Ex. 1001, 6:22-27.

Case IPR2014-00287

Patent 6,836,420 B1

E. Prior Art Relied Upon

Petitioner relies upon the following prior art references:

Shand, U.S. Patent No. 6,192,447 B1 (issued Feb. 20, 2001)
 (“Shand ’447,” Ex. 1006).

Runaldue, U.S. Patent No. 5,067,110 (issued Nov. 19, 1991)
 (“Runaldue ’110,” Ex. 1007).

Nels Vander Zanden, *Synthesis of Memories From Behavioral HDLs*,
 IEEE (1994) (“Vander Zanden,” Ex. 1003).

Peter Wohl, John Waicukauski, *Using Verilog Simulation Libraries
 For ATPG*, IEEE (1999) (“Wohl,” Ex. 1004).

XILINX SYNTHESIS TECHNOLOGY (XST) USER GUIDE, VERSION 3.1I,
 Xilinx, Inc. (2000) (“XST,” Ex. 1005).

F. The Alleged Grounds of Unpatentability

Petitioner contends that the challenged claims are unpatentable under
 35 U.S.C. §§ 102 and/or 103 on the following specific grounds.³

Reference(s)	Basis	Claims challenged
Vander Zanden (Ex. 1003)	§ 102	1, 2, 10-12, and 20
Wohl (Ex. 1004)	§ 102	1, 2, 10-12, and 20
XST (Ex. 1005)	§ 102	1, 2, 10-12, and 20
Vander Zanden and Shand (Ex. 1006)	§ 103	1-3, 10-13, and 20
Vander Zanden and Runaldue (Ex. 1007)	§ 103	1-3, 10-13, and 20

³ Petitioner supports its challenge with a Declaration by Mr. Ewald Detjens
 A.B., M.S. (“Detjens Decl.,” Ex. 1002).

Case IPR2014-00287

Patent 6,836,420 B1

XST and Shand	§ 103	1-3, 10-13, and 20
XST and Runaldue	§ 103	1-3, 10-13, and 20

II. CLAIM CONSTRUCTION

A. *Legal Standard*

Consistent with the statute and the legislative history of the Leahy-Smith America Invents Act (“AIA”), the Board will interpret claims of an unexpired patent using the broadest reasonable construction in light of the specification of the patent. *See* OFFICE PATENT TRIAL PRACTICE GUIDE, 77 Fed. Reg. 48,756, 48,766 (Aug. 14, 2012); 37 CFR § 42.100(b). Under the broadest reasonable construction standard, claims are to be given their broadest reasonable interpretation consistent with the specification, and the claim language should be read in light of the specification as it would be interpreted by one of ordinary skill in the art. *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004).

1. *Resettable memory*⁴

Both independent claims 1 and 11 include the step of, “inferring the existence of a *resettable memory*” (emphasis added). Petitioner proposes that under the broadest reasonable construction, a “resettable memory” should be construed as “a memory unit whose output value(s) can be cleared to a reset value, e.g., ‘0’, the memory unit comprising one or more storage cells.” Pet. 11 (emphasis and internal quotation marks omitted). Patent Owner contends that we should construe “resettable memory” in the same manner as in the

⁴ The parties at times use the term “resettable” with two t’s, while the claims of the ’262 patent use the term “resetable” with one t. In this decision we use the terms “resettable” and “resetable” interchangeably.

Case IPR2014-00287

Patent 6,836,420 B1

district court case, i.e. as “a memory unit with a plurality of cells, each cell associated with a unique address that provides access to that storage cell, where the memory unit can be reset to a particular value.” Prelim. Resp. 8-11; *see also* Ex. 1016, 9.

Although the parties provide competing constructions for “resettable memory,” neither party explains sufficiently why this term requires an express construction. On one hand, it is not clear from Petitioner’s argument and evidence that the specification of the ’262 patent broadly defines a resettable memory as “one or more storage cells.” *See* Pet. 12. On the other hand, Patent Owner does not argue in their Preliminary Response that Vander Zanden or Wohl fail to disclose a memory unit with a plurality of cells. *See* Prelim. Resp. 24-26, 30-33. Accordingly, we determine that no express interpretation of the term “resettable memory” is needed for this decision.

2. inferring

According to Petitioner, the term means “inferring or recognizing or identifying.” Pet. 14 (emphasis and internal quotation marks omitted). Patent Owner disagrees and argues that to the extent the term needs to be construed, the District Court’s construction giving the term its plain and ordinary meaning of “deducing” is the most appropriate interpretation. Prelim. Resp. 11-12.

Neither party points to, nor can we find, anything in the specification to indicate that “inferring” is used in a manner other than its ordinary and customary meaning. The plain meaning of “infer” is “to form an opinion or reach a conclusion through reasoning and information,” or “to convey an idea indirectly.” MERRIAM-WEBSTER ONLINE DICTIONARY,

Case IPR2014-00287

Patent 6,836,420 B1

<http://www.merriam-webster.com/thesaurus/infer> (last visited May 16, 2014). Synonyms are: conclude, decide, deduce, derive, extrapolate, gather, judge, make out, reason, and understand. *Id.* Mr. Detjens alleges that another term used for “inferring” is “recognizing.” Ex. 1002 ¶ 15. Based on the record before us, we are not persuaded that the meaning of “inferring” is better understood with reference to any one in particular of the parties’ proposed synonyms, or those from the dictionary. Our understanding is that the plain meaning of “inferring” includes: concluding, deciding, deducing, deriving, extrapolating, gathering, judging, making out, reasoning, understanding, and recognizing.

III. ANALYSIS

We turn now to Petitioner’s asserted grounds of unpatentability, and Patent Owner’s arguments in its preliminary response, to determine whether Petitioners have met the threshold standard of 35 U.S.C. § 314(a).

A. Claims 1-2, 10-12, and 20 – Anticipated by Vander Zanden (Ex. 1003)

On this record, Petitioner has established a reasonable likelihood of prevailing on its assertion that claims 1, 2, 10-12, and 20 are anticipated for the reasons explained below.

1. Overview of Vander Zanden

Vander Zanden discloses a method of creating descriptions of memory designs for use in computer emulation—synthesis—that differs from conventional techniques. Ex. 1003, 71. The method synthesizes a description of a memory, specifically a two-dimensional array *regFile*, apart from other logic and datapath elements as a behavioral description. *Id.* at

Case IPR2014-00287

Patent 6,836,420 B1

72. “This allows the designer to include the behavioral description of the memory with the rest of the HDL [Hardware Description Language] code, yet offers multiple architectural implementations for the memory without additional effort for the designer.” *Id.* at 71.

Figure 3 of Vander Zanden is reproduced below:

```
architecture regfilesyn_A of regfilesyn is
begin
  process(clk,rst)
    subtype addrType is integer range 0 to 15;
    type regFileType is array (addrType) of integer;
    variable regFile: regFileType;
  begin
    if rst='1' then
      out1 <= (others => '0');
    elsif clk'event and clk='1' then
      if s2='1' then
        out1 <= regFile(addr1) + regFile(addr2);
      else
        out1 <= regFile(addr3);
      end if;
      if s1='1' then
        regFile(writeAddr) := dataIn;
      end if;
    end if;
  end process;
end regfilesyn_A;
```

Figure 3: VHDL Description

Figure 3 of Vander Zanden illustrates an HDL model for a memory having an if-else statement that includes on one hand a data out, “out1” result of “0,” and on the other hand, a data out result of cell addresses, “addr1, 2, 3” (Ex. 1003, 73).

Figure 4 of Vander Zanden discloses a particular datapath design based on the HDL description of Figure 3 including two registers, an adder, and a multiplexor leading to the data out line (out1). *Id.* at 73. According to Vander Zanden the use of such a memory synthesis technique and datapath design reduces the size of the circuit design as well as operational delay. *Id.* at 74.

2. Analysis

Petitioner argues that Vander Zanden describes inferring a resettable memory from a behavioral or RTL level description, and also incorporating

Case IPR2014-00287

Patent 6,836,420 B1

a resettable memory design into a semiconductor circuit design as called for in claims 1 and 11. Pet. 15. Petitioner alleges a person of skill in the art would have understood that Vander Zanden's method of behavioral synthesis for an integrated circuit teaches inferring a memory from the two-dimensional array, i.e., *regFile*, within the behavioral description as shown in Figure 3. *Id.* (claim chart citing Ex. 1003, 72-73). Petitioner supports this position with the declaration of Mr. Detjens stating that by "[p]arsing a behavioral description and identifying '[a]ny two-dimensional array of bits' as 'a candidate for memory synthesis' [a] POSA would understand this parsing and identifying as inferring a memory from the behavioral description." Ex. 1002 ¶ 36 (citation omitted).

Relying on Mr. Detjens' declaration, Petitioner asserts that a person of skill in the art also would have understood that Vander Zanden teaches a semiconductor circuit design with a resettable memory where "Figure 4 shows a memory design in the form of a 16-bit register file cell and surrounding control logic, including a resettable flip-flop at the output of the memory." Ex. 1002 ¶ 39; *see* Pet. 16. Petitioner indicates that such a person of skill in the art "would understand the flip-flop 'DFF6' on the output of the memory to be a resettable flip-flop." Pet. 17 (citing Ex. 1002 ¶ 40).

Patent Owner argues that Vander Zanden never mentions a resettable memory and that Figure 4 shows the flip flop DFF6 on the output of the multiplexor, and therefore "does not provide reset functionality for the register file cell RGF1." Prelim. Resp. 22-23. Patent Owner further argues that Mr. Detjens, at paragraph 39, does not explicitly state that Vander Zanden's Figure 3 describes a resettable memory, nor clarifies how Figure 4 shows a resettable memory. Prelim. Resp. 23.

Case IPR2014-00287

Patent 6,836,420 B1

We are not persuaded by Patent Owner's arguments, because even though Vander Zanden and Mr. Detjens may not specifically state that the memory description in Figure 3 and circuit design shown in Figure 4 represent a "resettable memory," we are persuaded that both figures disclose resettable memory. The memory behavioral description in Figure 3 demonstrates that a reset value is applied to the "out1" variable when the "rst" variable equals "1." According to the Detjens declaration, a person of skill in the art "would understand that Vander Zanden's synthesis method identifies from the . . . clause in Figure 3 that a reset is being applied to the 'out1' variable when the 'rst' variable equals '1.'" Ex. 1002 ¶ 42.

We also understand from both parties' descriptions of Vander Zanden that register DFF6 in Figure 4 apparently applies a reset to the output "out1" value subsequent to the adder "ADD1" and multiplexer. Pet. 17-18; Prelim. Resp. 22. Patent Owner asserts that this is structurally and functionally different from the present invention that essentially applies a reset value prior to the multiplexer. Prelim. Resp. 22-24. Patent Owner's position that "the register DFF6 does not provide reset functionality for the register file cell RGF1" is not persuasive because, similar to the '420 patent, any downstream device or program from DFF6 would see an "effective" memory unit output "out1" that is either a reset value, or an actual data value from an address in RGF1. Prelim. Resp. 23; *see also* Ex. 1001, 3:13-20. Moreover, neither claim 1 nor claim 11 recites any particular resettable memory design structure that excludes a circuit component from applying a reset value following the multiplexer.

Patent Owner suggests that Vander Zanden does not describe the "operational flow" of a resettable memory as recited in claims 10 and 20

Case IPR2014-00287

Patent 6,836,420 B1

because the “out1” variable is not disclosed as part of the memory, RGF1, and because Vander Zanden does not explicitly state that the method is “inferring” or “identifying” a resettable memory. Prelim. Resp. 26-27. We are not persuaded by these arguments because Patent Owner does not articulate why the output “out1” of register DFF6 is structurally or functionally different from the claimed invention, and does not explain why the register DFF6 and “out1” are not part of the “operational flow” of a resettable memory. *Id.* at 27. Further, the fact that Vander Zanden does not explicitly use the terms “inferring” or “identifying” does not explain why downstream devices and programs from DFF6 would not infer, recognize, or deduce, that the memory was resettable, given that the output “out1” could be a reset value. *Id.*

Patent Owner does not address the respective dependent claims, apart from claims 10 and 20 as discussed above. *See* Prelim. Resp. 27. We have considered Petitioner’s arguments and evidence concerning the remaining dependent claims and, on this record, are persuaded of a reasonable likelihood of Petitioner’s prevailing as to them, as well. For the above reasons, and based on the record before us, Petitioner has established a reasonable likelihood of prevailing on the ground of unpatentability of claims 1, 2, 10-12, and 20 as anticipated by Vander Zanden under 35 U.S.C. § 102(b).

B. Claims 1-2, 10-12, and 20 – Anticipated by Wohl (Ex. 1004)

On this record, Petitioner has established a reasonable likelihood of prevailing on its assertion that claims 1, 2, 10-12, and 20 are anticipated for the reasons explained below.

Case IPR2014-00287

Patent 6,836,420 B1

1. Overview of Wohl

Wohl describes an improvement in integrated circuit design flow that more efficiently uses a single Verilog simulation library for both simulation and ATPG (“Automatic Test Pattern Generator”) model building. Ex. 1004, 1011. Wohl explains that using a single library in the netlist reader and model builder of an ATPG tool eliminates the necessity for coding and verifying a separate ATPG library, and additionally simplifies debugging test problems. *Id.*

Wohl specifically discloses an ATPG model builder for converting a behavioral Verilog description, for example of a RAM including a reset line and read/write ports, into a resulting ATPG model test pattern generator. *Id.* at 1017, Figs. 6-7. Wohl discusses a methodology which rewrites, or recodes, the Verilog RAM description into a simplified form that is sufficiently simple for automatic processing and displays the resulting ATPG RAM model in a schematic viewer. *Id.* By way of example, the RAM ATPG model shown in Wohl’s Figure 7 appears as a high-level structural circuit design for a test pattern generator simulated by the behavioral Verilog description shown in Figure 6.

2. Analysis

Petitioner asserts that Wohl’s Figure 6 discloses a resettable memory with a RAM described in behavioral Verilog including a reset line. Pet. 20. Petitioner contends that Wohl uses an ATPG model builder to convert the behavioral Verilog description into a RAM ATPG model, and alleges that a person of ordinary skill in the art “would understand this conversion to require inferring, or recognizing, the existence of a resettable memory from the behavioral Verilog description.” *Id.* (citing Ex. 1002 ¶¶ 46-47).

Case IPR2014-00287

Patent 6,836,420 B1

Directing our attention to Figure 7, Petitioner further contends that the RAM ATPG model depicts a design for a semi-conductor circuit. *Id.* at 20-21.

Patent Owner argues that the RAM ATPG model, disclosed by Wohl's Figure 7, is a simplified circuit model intended only for generating test patterns, which omits most of the functionality of the memory, and therefore "[is] not a design for a 'semiconductor circuit' and the RAM ATPG model[] cannot be used to manufacture a semiconductor circuit." Prelim. Resp. 29. Patent Owner argues that the simple RAM ATPG model discloses nothing in regard to a larger memory model or circuit. *Id.* at 30.

We are not persuaded by this argument because the claims impose no design size or memory characteristic limitations on the claimed semiconductor circuit. The RAM ATPG model disclosed in Wohl's Figure 7 is a schematic representation of a basic structural circuit with read/write ports, memory addresses and outputs 2d[7]. *See* Ex. 1004, 1017. Patent Owner's position that the RAM ATPG model of Figure 7 is a test pattern generator does not persuade us that it is not representative of a semiconductor circuit. Independent claims 1 and 11 recite only "a semiconductor circuit," which does not exclude a test pattern generator.

Patent Owner asserts that Mr. Detjens tacitly admits that the ATPG model shown in Wohl's Figure 7 is not a semiconductor circuit. Prelim. Resp. 30. Specifically, Patent Owner argues that because Mr. Detjens states that Figure 7 is an ATPG model for generating test patterns, the ATPG model cannot therefore also be a semiconductor design. *Id.* at 31 (citing Ex. 1002 ¶47). We are not persuaded that Mr. Detjens has made such an admission by acknowledging that Wohl's circuit schematic shown in Figure 7 is a test pattern generator. Mr. Detjens states that "figure 7 shows the

Case IPR2014-00287
Patent 6,836,420 B1

ATPG model built as displayed by the schematic viewer.” Ex. 1002 ¶ 46 (citing Ex. 1004, 1017). The schematic shown in Figure 7 appears to be at least representative of a semiconductor circuit. On the record before us, Patent Owner has not persuaded us that there is any fundamental difference between the schematic representation of a RAM ATPG RAM model in Wohl, and a basic semiconductor circuit design.

Patent Owner further asserts that the ATPG RAM model, as it generates test patterns, “is not for producing semiconductor circuits.” Prelim. Resp. 30. As discussed above, we are not persuaded that the RAM ATPG model fails to disclose a semiconductor circuit. To the extent Patent Owner further argues the claims are drawn to a functional step of “producing” a circuit, we note that independent claims 1 and 11, recite the step of “incorporating a resettable memory design into a design for said semiconductor circuit,” not “producing” a semiconductor circuit.

Patent Owner next argues that Wohl’s strategy to manually rewrite i.e. recode, the Verilog description does not “infer [] a resettable memory” as claimed. Prelim. Resp. 32. Specifically, Patent Owner asserts that “Figure 6 shows only the simplified description of memory that must be adopted through recoding to allow use of the ATPG tool.” *Id.* We are not persuaded by this position because Figure 6 discloses an “if-else” reset statement from which the ATPG RAM model of Figure 7 appears to infer, recognize, or deduce that the RAM (memory) is resettable, with “set” and “reset” lines in the ATPG RAM model.

Patent Owner next contends because Wohl fails to disclose inferring memory, it cannot then teach “identifying within an operational flow of said description that a reset is being applied to a variable” as called for in claims

Case IPR2014-00287

Patent 6,836,420 B1

10 and 20. Prelim. Resp. 33. We are not persuaded by this argument because the conversion from the behavioral Verilog description interprets the “if-else” statement of Figure 6 such that a reset is applied to mymem[i] variable, “if” the reset signal goes to 1.

With respect to claims 11, 12, and 20, Patent Owner asserts that Wohl also does not disclose “a machine readable medium” because Wohl explains that the Verilog description is manually recoded into a simplified form for processing. Prelim. Resp. 34. Petitioner argues that Wohl teaches carrying out the steps of claims 11, 12 and 20 on a computer “through use of the ‘netlist reader and model builder of a commercial ATPG tool,’” and “automatic” operations. Pet. 22 (citing Ex. 1004, 1011). Petitioner contends that this disclosure “teaches a machine readable medium having stored thereon a sequence of instructions which, when executed by a digital processing system, cause said system to perform the method steps.” *Id.* at 22-23 (citing Ex. 1002 ¶ 50).

Wohl states that the described methodology and ATPG model builder functions occur in the context of faster CPU (central processing unit) parsing and translating times. Ex. 1004, 1020. Although Wohl does not specifically state that the methodology is carried out by a “machine readable medium,” we are not persuaded by Patent Owner’s position that Wohl’s simulation implemented in the netlist reader and model builder functions is accomplished without instructions stored on a machine readable medium and without using a CPU and stored instructions to implement the recoded Verilog description and conversion. *See* Prelim. Resp. 34.

We have considered Petitioner’s arguments and evidence concerning the remaining dependent claim 2 and, on this record, are persuaded of a

Case IPR2014-00287

Patent 6,836,420 B1

reasonable likelihood of Petitioner's prevailing as to it, as well. For the above reasons, and based on the record before us, Petitioner has established a reasonable likelihood of prevailing on the ground of unpatentability of claims 1, 2, 10-12, and 20 as anticipated by Wohl under 35 U.S.C. § 102(b).

C. Claims 1-3, 10-13, and 20 – Obvious over Vander Zanden and Shand, or Vander Zanden and Runaldue

On this record, Petitioner has established a reasonable likelihood of prevailing on its assertion that claims 1-3, 10-13, and 20 are obvious for the reasons explained below.

Shand discloses for example in Figure 1 a non-resettable RAM 100 and a reset register 140 with corresponding bits “m” to respective memory locations in RAM 100. Ex. 1006, 2:31-66. When a reset register bit value is valid, “the reset value from register 140 is produced on line 130 via lines 162 and 161, instead of the actual data stored at the addresses memory location.” *Id.* at 2:64-66.

For its part, Runaldue discloses a memory circuit with a zero detect logic to determine if a row of a memory array is to be treated as reset to a zero state, or preserved in a non-zero state according to corresponding tag-memory cells. Ex. 1007, 2:1-16. Runaldue explains that, “[t]he tag-memory cells are resettable to a zero state to indicate that the associated row of the memory array is to be treated as all being in the zero state. This is to be done even though the actual contents of the memory are not all zeros.” *Id.* at 2:11-16.

Petitioner argues that a person of skill in the art would understand that both Shand and Runaldue compliment Vander Zanden, since both Shand and Runaldue relate specifically to resettable memories, and each describes

Case IPR2014-00287

Patent 6,836,420 B1

circuit designs for a memory that can be quickly reset without substantially increasing the number of circuit elements. Pet. 31, 40-41.

Petitioner supports the combination of Vander Zanden and the secondary references with Mr. Detjens' declaration, who testifies that where Vander Zanden's method identifies all the characteristics of a memory, including reset, to incorporate the memory into a gate-level netlist, a person of skill in the art would look for appropriate resettable memory circuit designs such as described in Shand and Runaldue. Ex. 1002 ¶¶ 66, 69. Mr. Detjens asserts that when designing a memory a person of ordinary skill would have considered, among other factors, design area, power consumption, speed, available gates and macro cells of the target library, and would have naturally looked to Shand which, like the '420 patent, addresses such issues with "creative design approaches that allowed for synthesis of a resettable memory design using off-the-shelf components." *Id.* ¶¶ 67-68. Mr. Detjens contends one of ordinary skill would have also sought to undertake Vander Zanden's memory synthesis using Runaldue's memory circuit design because Runaldue's efficient and fast resettable memory design has compatible characteristics with Vander Zanden's memory synthesis and uses "'tag-memory cells [that] are resettable to a zero state,' and a non-resettable 'memory array,'" which were readily available off-the-shelf components. *Id.* ¶ 69 (alteration in original) (citing Ex. 1008, 2:12, 4:17-23, 4:41-44, and Figs. 5, 7).

In its arguments discussing the combination of each of Shand and Runaldue with Vander Zanden, Patent Owner argues, as in arguing the anticipation issue above, that Vander Zanden does not describe resettable memories or inferring resettable memories, and that Petitioner is only

Case IPR2014-00287

Patent 6,836,420 B1

speculating as to how a person of skill in the art would understand Vander Zanden. Prelim. Resp. 48, 50. Petitioner however relies upon Vander Zanden for teaching memory synthesis and the step of inferring resettable memories. Pet. 31. For the same reasons discussed above Patent Owner's argument is not persuasive. *See supra* section III.A.2.

Patent Owner further argues that neither Shand nor Runalduie discloses an HDL or RTL description that infers a resettable memory. Prelim. Resp. 49, 51. This argument is not persuasive because, as above, Petitioner relies upon Vander Zanden, not Shand, for these elements of claims 1 and 11. *See supra* section III.A.2.

Patent Owner also contends that neither the combination of Vander Zanden and Shand, nor Vander Zanden and Runalduie, discloses the limitations of claims 10 and 20 because the secondary references do not disclose inferring a resettable memory. Prelim. Resp. 49, 51. This argument is not persuasive, again, because Petitioner relies upon Vander Zanden, not Shand, for these elements of claims 10 and 20. *See supra* section III.A.2.

Patent Owner does not discuss any deficiencies in Petitioner's position with respect to claims 2, 3, 12, 13, and 20. For the reasons discussed, and based on the record before us, Petitioner has established a reasonable likelihood of prevailing on the ground of unpatentability of claims 1-3, 10-13, and 20 as obvious over Vander Zanden and Shand, or Vander Zanden and Runalduie.

D. Additional Grounds

The alleged grounds of unpatentability that claims 1, 2, 10-12, and 20 would have been anticipated by XST, and that claims 1-3, 10-13, and 20 would have been obvious over XST and Shand or XST and Runalduie, are

Case IPR2014-00287

Patent 6,836,420 B1

redundant in light of the grounds on which we institute review for the same claims. Accordingly, we exercise our discretion by not instituting review on these additional grounds. *See* 37 C.F.R. § 42.108(a).

IV. SUMMARY

For the foregoing reasons, we determine that the information presented in the petition establishes a reasonable likelihood that Petitioner would prevail on at least one alleged ground of unpatentability with respect to each of claims 1-3, 10-13, and 20 of the '420 patent.

The Board has not made a final determination on the patentability of any challenged claims.

V. ORDER

For the reasons given, it is

ORDERED that the petition is *granted* as to all the challenged claims of the '420 patent on the following grounds:

1. Claims 1, 2, 10-12, and 20 as anticipated by Vander Zanden;
2. Claims 1, 2, 10-12, and 20 as anticipated by Wohl;
3. Claims 1-3, 10-13, and 20 as obvious over Vander Zanden and Shand; and
4. Claims 1-3, 10-13, and 20 as obvious over Vander Zanden and Runaldue;

FURTHER ORDERED that all other grounds raised in the petition are denied because they are redundant in light of the grounds on the basis of which we institute an *inter partes* review for the same claims;

Case IPR2014-00287

Patent 6,836,420 B1

FURTHER ORDERED that pursuant to 35 U.S.C. § 314(a) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial on the grounds of unpatentability authorized above; the trial commences on the entry date of this decision.

Case IPR2014-00287

Patent 6,836,420 B1

For PETITIONER:

John D. Vandenberg
Andrew M. Mason
KLARQUIST SPARKMAN, LLP
john.vandenberg@klarquist.com
andrew.mason@klarquist.com

For PATENT OWNER:

William H. Wright
Travis Jensen
ORRICK, HERRINGTON, &
SUTCLIFFE LLP
wwright@orrick.com
tjensen@orrick.com

**Final Written Decision,
Dated June 11, 2015 (Paper No. 31)**

Trials@uspto.gov
571.272.7822

Paper 31
Entered: June 11, 2015

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MENTOR GRAPHICS CORPORATION,
Petitioner,

v.

SYNOPSYS, INC.,
Patent Owner.

Case IPR2014-00287
Patent 6,836,420 B1

Before JENNIFER S. BISK, SCOTT A. DANIELS, and
PHILIP J. HOFFMANN, *Administrative Patent Judges*.

DANIELS, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

IPR2014-00287
Patent 6,836,420

I. INTRODUCTION

A. *Background*

Mentor Graphics Corporation (“Petitioner”) filed a Petition to institute an *inter partes* review of claims 1–3, 10–13, and 20 of U.S. Patent No. 6,836,420 B1 (“the ’420 patent”). Paper 6 (“Pet.”).¹ We instituted trial for claims 1–3, 10–13, and 20 of the ’420 patent on certain grounds of unpatentability alleged in the Petition. Paper 9 (“Decision to Institute” or “Inst. Dec.”).

After institution of trial, on September 17, 2014, Patent Owner, Synopsys, Inc., (“Patent Owner”), filed a Patent Owner Response (“PO Resp.”), along with a Declaration by Patent Owner’s Declarant, Dr. Brad Hutchings (“Hutchings Declaration”). Subsequently, Petitioner filed a Reply (“Reply”) on December 12, 2014 and a Declaration in support of the Reply by Petitioner’s Declarant, Mr. Edward Detjens (“Reply Declaration”). Paper 22.

A hearing for IPR2014-00287 was held on March 10, 2015. The transcript of the hearing has been entered into the record. Paper 30 (“Tr.”).

We have jurisdiction under 35 U.S.C. § 6(c). This final written decision is issued pursuant to 35 U.S.C. § 318(a).

Petitioner has shown by a preponderance of the evidence that claims 1–3, 10–13, and 20 of the ’420 patent are unpatentable based on the combination of Vander Zanden and Shand. Petitioner has not shown that the challenged claims are unpatentable over any of the other proposed grounds.

¹ We refer to the corrected Petition filed January 15, 2014.

Case IPR2014-00287

Patent 6,836,420

B. The '420 Patent

The '420 patent (Ex. 1001) generally relates to memory circuit design and specifically, a method and corresponding digital circuit design for resetable memory. Ex. 1001, 1:7–10. The '420 patent states, as a matter of background, that the circuitry for a conventional resetable memory unit is complicated and expensive because “each n wide storage cell is implemented with resetable flip-flops that are individually accessed via complicated multiplexing and control circuitry.” *Id.* at 1:65–67. According to the patent, such a resetable memory unit is relatively slow and consumes more silicon surface area than a non-resetable memory unit. *Id.* at 2:3–6.

A solution to these challenges, proposed by the '420 patent, is designing resetable memory 220 as a combination of memory unit without reset 201, and memory unit with reset 205. *Id.* at 1:61–2:2, Fig. 2A. Figure 2A of the '420 patent, illustrating resetable memory 220, is reproduced below.

Case IPR2014-00287

Patent 6,836,420

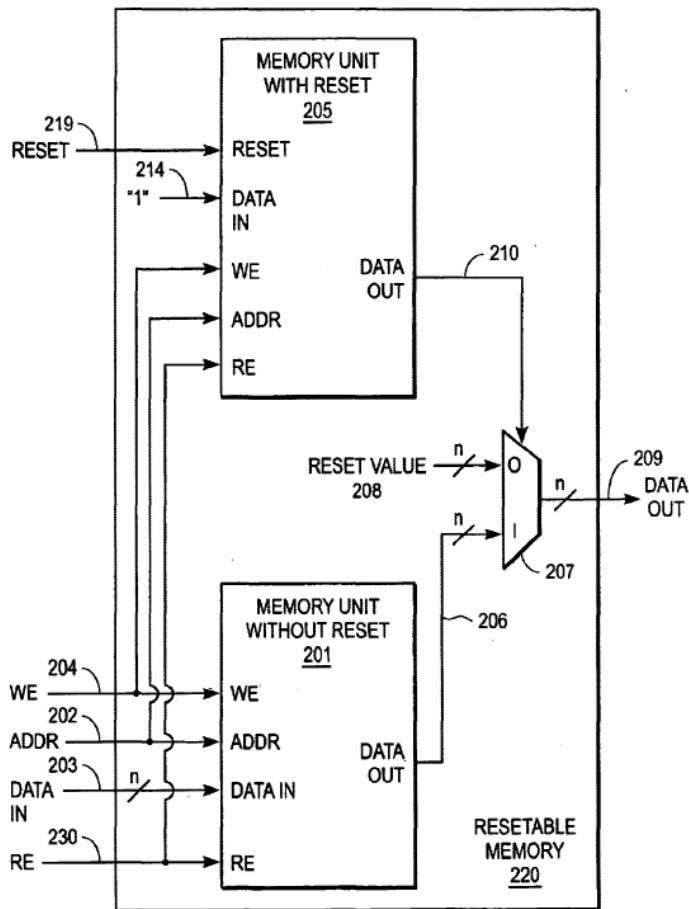


FIG. 2A

As depicted by Figure 2A, above, both memory unit without reset 201 and memory unit with reset 205 are connected to data out line 209 via multiplexor 207. Memory unit without reset 201 is larger in that it has a greater data width than memory unit with reset 205. *Id.* at 3:3–10. The '420 patent describes the smaller resettable memory having a cell word size less than the cell word size of the non-resettable memory. For example, the cell word size of the resettable memory is only one bit wide, so that the memory output from a cell is either a "1" or "0." *Id.* at 3:8–10. As explained below, the purpose of the combination is that the smaller memory with reset 205

Case IPR2014-00287

Patent 6,836,420

“cost effectively disguises the inability of the larger memory 201 to reset its cells.” *Id.* at 3:12–13.

Resettable memory 220 operates generally as follows: data is provided through data in line 203 to memory without reset 201, and is stored in a particular cell within the memory. *See id.* at Fig. 2A. The '420 patent refers to data called from memory without reset 201 as “actual memory unit data output 206.” *Id.* at 3:26–30. Memory unit with reset 205 has a corresponding cell, in which a “1” from data in 214 is stored whenever write enable (WE) 204 of resettable memory 220 is activated. *Id.* at 4:1–3. The “1” from the memory unit with reset 205 instructs multiplexer 207 to output the actual memory unit data output 206 from memory 201 to data out line 209. *Id.* at 4:30–35. If memory unit with reset 205 is reset, a reset value, for example “0,” is stored in memory unit 205, output to data out 210, and received by Multiplexer 207. In this circumstance, reset value 208 is output to data out line 209. *Id.* at 3:51–54. If no new data has been written to the particular cell in memory unit without reset 201 since its last reset, the multiplexer will continue to output the reset value “0,”² which remains in its storage cell until WE 204 line is activated again. *Id.* at 4:1–3. Accordingly, when a new value is written to the particular cell in memory unit without reset 201, the “0” will be over written with a “1” and the new value will be output by multiplexer 207 to data out line 209. *Id.* at 4:30–35. The '420 patent explains that “[i]n this manner, the circuit of FIG. 2[A] emulates the

² The reset may be a value besides “0,” for instance FIG. 2B discloses an embodiment in which “a reset value function circuit 230 may be inserted between (and coupled to) the reset value 208 input of the multiplexer 207 and the address input 202 of the resettable memory 220.” Ex. 1001, 4:67–5:3.

Case IPR2014-00287

Patent 6,836,420

behavior of a memory unit having the storage capacity of memory unit 201 but also having reset capability.” *Id.* at 4:46–48.

In other words, the memory unit with reset 205 effectively replicates the behavior of a resettable memory by providing “0” or another reset value, if there has been a reset of a corresponding cell in memory unit with reset 205, unless and until new data is written to memory 201. *See id.* at 4:46–54.

In addition to the resettable memory circuit embodiment of Figure 2A discussed above, the ’420 patent describes a circuit design methodology inferring the use of a resettable memory from the behavioral level, or RTL (register-transfer level) description of the memory. *Id.* at 5:32–34. Rather than describe the specific circuit hardware, RTL level description describes the memory circuit in terms of its function, or operational flow, including the characteristic of the memory that it is resettable. *Id.* at 5:36–44. A software design tool, by way of example, which facilitates circuit design, infers from the RTL level description that a reset condition is being applied to at least one specific variable, and the software design tool can incorporate a resettable memory into the designer’s circuit design.³ *Id.* at 6:27–29; *see* Fig. 4.

C. Illustrative Claims

Of the challenged claims, the independent claims are 1 and 11. Each of dependent claims 2, 3, and 10 depends directly from claim 1. Each of

³ The ’420 patent states that “[t]he automatic inference can be accomplished, for example, by configuring the design tool to recognize from the operational flow of the circuit that: 1) some type of reset is being applied to the stored data values within the circuit; and 2) the stored data values are being changed to some type of reset value in response.” Ex. 1001, 6:22–27.

Case IPR2014-00287

Patent 6,836,420

dependent claims 12, 13, and 20 depends directly from claim 11. Claims 1 and 11 illustrate the claimed subject matter and are reproduced below:

1. A method, comprising:

- a) inferring the existence of a resetable memory from a behavioral or RTL level description of a semiconductor circuit; and
- b) incorporating a resetable memory design into a design for said semiconductor circuit.

11. A machine readable medium having stored thereon a sequence of instructions which, when executed by a digital processing system, cause said system to perform a method, said method, comprising:

- a) inferring the existence of a resetable memory from a behavioral or RTL level description of a semiconductor circuit; and
- b) incorporating a resetable memory design into a design for said semiconductor circuit.

D. The Prior Art References Supporting Alleged Unpatentability

Petitioner relies upon the following prior art references:

Shand, U.S. Patent No. 6,192,447 B1 (issued Feb. 20, 2001)
("Shand '447," Ex. 1006).

Runaldue, U.S. Patent No. 5,067,110 (issued Nov. 19, 1991)
("Runaldue '110," Ex. 1007).

Nels Vander Zanden, *Synthesis of Memories From Behavioral HDLs*, IEEE (1994) ("Vander Zanden," Ex. 1003).

Peter Wohl & John Waicukauski, *Using Verilog Simulation Libraries For ATPG*, IEEE (1999) ("Wohl," Ex. 1004).

XILINX SYNTHESIS TECHNOLOGY (XST) USER GUIDE, VERSION 3.1I, Xilinx, Inc. (2000) ("XST," Ex. 1005).

Case IPR2014-00287

Patent 6,836,420

E. The Pending Grounds of Unpatentability

Reference(s)	Basis	Claims challenged
Vander Zanden (Ex. 1003)	§ 102	1, 2, 10–12, and 20
Wohl (Ex. 1004)	§ 102	1, 2, 10–12, and 20
Vander Zanden and Shand (Ex. 1006)	§ 103	1–3, 10–13, and 20
Vander Zanden and Runaldue (Ex. 1007)	§ 103	1–3, 10–13, and 20

Petitioner supports its challenge with a Declaration by Mr. Ewald Detjens A.B., M.S. (“Detjens Decl.,” Ex. 1002).

II. CLAIM CONSTRUCTION

A. *Legal Standard*

Consistent with the statute and the legislative history of the Leahy-Smith America Invents Act (“AIA”), Public Law 112-29, 125 Stat. 284 (September 16, 2011), the Board will interpret claims of an unexpired patent using the broadest reasonable construction in light of the specification of the patent. *See* Office Patent Trial Practice Guide, 77 Fed. Reg. 48,756, 48,766 (Aug. 14, 2012); 37 C.F.R. § 42.100(b). Under the broadest reasonable construction standard, claims are to be given their broadest reasonable interpretation consistent with the specification, and the claim language should be read in light of the specification as it would be interpreted by one of ordinary skill in the art. *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004), *see In re Cuozzo Speed Techs., LLC*, 778 F.3d 1271, 1279–83 (Fed. Cir. 2015). Also, we must be careful not to read a particular embodiment appearing in the written description into the claim, if the claim language is broader than the embodiment. *See In re Van Geuns*,

Case IPR2014-00287

Patent 6,836,420

988 F.2d 1181, 1184 (Fed. Cir. 1993) (“[L]imitations are not to be read into the claims from the specification.”).

B. Overview of the Parties’ Positions

1. Inferring and Incorporating

In the Decision to Institute, we provided an interpretation for “inferring” in accordance with its plain meaning, including: concluding, deciding, deducing, deriving, extrapolating, gathering, judging, making out, reasoning, understanding, and recognizing. Inst. Dec. 9–10. We did not provide a construction for any other terms.

Patent Owner’s position is that the word “deducing” comes closer to the meaning of “inferring” than the words “recognizing” or “identifying” as proposed by Petitioner. PO Resp. 24, *see also* Pet. 14. Our construction includes from its plain meaning, a variety of words (including: deducing recognizing, and identifying) that, depending on context, provide additional understanding of the word “inferring.” Inst. Dec. 9–10. A plain meaning of the word “incorporating” is “to unite or work into something already existent so as to form an indistinguishable whole.” *Incorporate*, Merriam-Webster Online Dictionary, <http://www.merriam-webster.com/dictionary/incorporate> (last visited May 28, 2015).

For these words, we do not consider the proffered constructions to provide any clarity over the term itself. Consequently, we simply are not persuaded by either of the parties contentions and interpretations that under the broadest reasonable interpretation, in the context of the Specification and claims, that these words should be construed with respect to only the particular definitions or meanings ascribed to them by either party.

Case IPR2014-00287

Patent 6,836,420

2. *Memory*

Based on the parties' positions and arguments in the Petition, Patent Owner's Response, Petitioner's Reply, as well as at the oral hearing, with respect to Vander Zanden, the base reference in each ground, we determine that the word "memory" should be construed explicitly.

Neither party provides a construction for "memory," however, the '420 patent describes a "memory unit" in the Background section of the Specification as,

having a plurality of storage cells (or simply, "cells"). Associated with each cell is a unique address that provides access to the location of a particular storage cell. Each storage cell has the capacity to store "n" bits (where n is an integer greater than or equal to one). The n bits may be collectively referred to as a word of data.

Ex. 1001, 1:15–20. The '420 patent explains that for a memory, from an input perspective, a word of data (i.e. "n" bits of data) is written to a specific cell address in the memory unit, and from an output perspective, the word of data is retrieved from a provided cell address and "the word of data is presented at the data output bus." *Id.* at 21–38. Thus, the '420 patent provides certain structural and functional characteristics that provide a basis for defining "memory." Moreover, a person of ordinary skill in the art at the time of the invention would have understood based on these characteristics that a memory unit in the '420 patent either presents a word of data from a specified address at the output in a "read" command, or overwrites old data with a new word of data at a specified address in a "write" command. Hutchings Decl. ¶ 21. ("One or more inputs control whether the current contents of a memory word should be presented at the output (referred to as

Case IPR2014-00287

Patent 6,836,420

a ‘read’) or whether the current contents should be overwritten and updated to contain a new value (referred to as a ‘write’).”) Based on the Specification and evidence before us, we construe “memory” as a device for storing data having a plurality of cells, each cell having a unique address for storing data, where data is written to a cell during a write function, and, during a read function data is retrieved from a cell and presented at a memory output.

3. *Resettable Memory*

Petitioner provides contentions regarding the broadest reasonable construction of “resettable memory.” Pet. 10–14. Specifically, Petitioner contends that because the ’420 patent describes that the memory unit’s cells are not actually reset, but only appear to be reset to downstream components, “resettable memory” is “a memory unit whose *output value(s)* can be cleared to a reset value, e.g., ‘0’, the memory unit comprising one or more storage cells.” *Id.* at 11 (emphasis added). Patent Owner argues that Petitioner’s construction is overly broad, and that the Specification of the ’420 patent repeatedly describes that a “resettable memory” “must output a reset value for a given memory cell (following a reset) *until new data is written into that cell.*” PO Resp. 12–13 (emphasis added). Petitioner counters that the construction of this term should not be conditioned on future data writes. Reply 6–7.

A claim construction analysis begins with, and is centered on, the claim language itself. *See Interactive Gift Express, Inc. v. Compuserve, Inc.*, 256 F.3d 1323, 1331 (Fed. Cir. 2001). Nevertheless, claims must be read in view of the specification of which they are a part.’ *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007), *Phillips v. AWH Corp.*, 415 F.3d

Case IPR2014-00287

Patent 6,836,420

1303, 1315, (Fed.Cir. 2005) (en banc). The specification is the single best guide to the meaning of a disputed term. *Id.* Turning to the language of claim 1, the claimed method includes two steps “inferring the existence of a resetable memory,” and “incorporating a resetable memory design,” but the claim language does not, in any detail, explicate what the “resetable memory” itself, is, or does. Ex. 1001, claim 1.

The Specification of the ’420 patent states from a structural standpoint that “a resetable memory is described that includes a memory without reset capability.” Ex. 1001, 2:51–52. The purpose of including a “memory without reset” in the overall resetable memory, the Specification explains, is to be less expensive and less complex than an actual “memory with reset” because “often, the integration of circuitry for resetting the cell word values of the memory unit **101** is too expensive and/or complicated to implement . . . having noticeably slower performance (and that consumes more silicon surface area).” *Id.* at 1:61–2:6. The “memory without reset” portion of the overall resetable memory is smaller and more efficient, but unable to reset its memory cells. *Id.* at 3:12–13.

As discussed in further detail below, we are not apprised of a sufficient reason to read “memory without reset” into the claims. Though understanding the claim language may be aided by the explanations contained in the written description, it is important not to import into a claim limitations that are not a part of the claim. *Superguide Corp. v. DirecTV Enters., Inc.*, 358 F.3d 870, 875 (Fed. Cir. 2004). Nevertheless, as described explicitly in the ’420 Specification, when reset, the overall “resetable memory” must be able to output a reset value, despite retention of what is

Case IPR2014-00287

Patent 6,836,420

essentially stored, old data in a particular memory cell of the resettable memory. *See* Ex. 1001, 3:11–23.

Patent Owner alleges that the Specification consistently discloses that “a ‘resettable memory’ must output a reset value for a given memory cell (following a reset) until new data is written into that cell.” PO Resp. 13–17 (citing Ex. 1001, 3:40–58, 4:17–21, 47–54, 7:12–31, Figs. 2A, 2B, 3, 5, 6; Hutchings Decl. ¶¶ 31, 36–40). Dr. Hutchings testifies that a person of ordinary skill in the art reading the ’420 patent “would understand that one defining characteristic of a resettable memory is that, following a reset, it must output a reset value for a given memory cell until new data is written into that cell.” Hutchings Decl. ¶ 31. It is further stated by Dr. Hutchings that in observing the operating flow of the invention from Figure 3 of the ’420 patent, the implementation of the described invention relies explicitly on the methodology that “*a reset is asserted 301. This causes the reset value 208 (in FIGS. 2A and 2B) to be provided 302 as the effective memory cell output-until the cell is written to.*” *Id.* ¶ 35 (citing Ex. 1001, 4:56–61).

Our review of the Specification indicates that the “resettable memory” is consistently defined as “reset” in each embodiment to output the reset value (as opposed to the stored, old, data in the memory without reset) until a new data value is written in a particular memory cell. Ex. 1001, 3:50–56 (“That is, after the resettable memory **220** has been ‘reset’, any attempt to read a data word from a particular cell within the memory unit without reset **201** will produce the reset value **208** at the memory unit data output **209**. This functional behavior continues for each cell until a particular cell is written to.”) We are mindful that we should not ordinarily rely on the preferred embodiments alone as representing the entire scope of the claimed

Case IPR2014-00287

Patent 6,836,420

invention. *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1370 (Fed. Cir. 2002); *see also Rexnord Corp. v. Laitram Corp.*, 274 F.3d 1336, 1344 (Fed. Cir. 2001) (emphasizing that the scope of a claim term often covers more than the embodiments disclosed in the specification and that a patent applicant need not describe “in the specification every conceivable and possible future embodiment of his invention”). However, Patent Owner’s evidence from the ’420 patent unambiguously describes how all the embodiments include this functionality. To construe this term as Petitioner proposes as simply “a memory unit whose output value(s) can be cleared to a reset value, e.g., ‘0’,” ignores the explicit behavior and the methodology of a “resettable memory” as described in the Specification. Pet. 11. Moreover, Petitioner’s proposed construction relies in part on a statement from the Background of the ’420 patent explaining that in the context of known resettable memories “[a] reset function effectively ‘clears’ the memory unit’s cell word values to some ‘reset’ value.” *Id.* (citing Ex. 1001, 1:59–60.) This statement, however, relates to conventionally clearing the memory unit’s cell value, not the “effective” output discussed in context of the inventive resettable memory described later in the detailed description. *See* Ex. 1001, 3:24–26.

Petitioner’s apparent position that the word “effectively” ties this phrase discussing known circuitry for resetting cell word values in a memory unit, to the “‘effective’ memory unit output as observed by the downstream circuitry,” discussed later in the Specification is not persuasive. *Id.* at 3:32–34. Indeed, supported as it is by the Background of the invention, Petitioner’s incorrect construction would encompass the example described in the Background section of the Specification, i.e. the known method using

Case IPR2014-00287

Patent 6,836,420

a resetable flip-flop for clearing each storage cell in the memory. *See id.* at 1:63–2:6. This is not a reasonable interpretation owing to the detailed description and more specific characterization of the described “resetable memory,” contrasted against the known resetable memory described in the Background section of the ’420 patent. Thus, we are persuaded that one of ordinary skill in the art, would understand from the Specification of the ’420 patent that a “resetable memory,” as claimed, is a memory, as construed above, that outputs a reset value until new data is written into the memory.

4. *Resetable Memory Design*

Claims 1 and 11, in paragraphs b), both recite a “resetable memory design.” In the context of both the method recited in claim 1, and the machine readable medium recited in claim 11, paragraph b) reads:

b) incorporating a resetable memory design into a design for said semiconductor circuit.

On its face, the predicate in this clause uses the word “design” in the context of a circuit, i.e. a semiconductor circuit design. Reviewing the Specification, the term “design” is similarly used in almost every instance to refer to a circuit or semiconductor circuit. For example, the Specification explains the benefits of implementing the resetable memory in a circuit design, where:

[a] further utility of the approaches discussed above is the ease at which a memory having reset may be incorporated into a *designer’s circuit design* . . . [f]or example, *semiconductor circuits* are typically designed with a particular semiconductor manufacturing process (i.e., a “foundry”) in mind. Usually, the foundry supplies models of basic building blocks (e.g., logic gates, memory units, etc.) from which a *semiconductor chip design* can be constructed.

Case IPR2014-00287

Patent 6,836,420

Ex. 1001, 5:9–18 (emphasis added).

From the plain meaning of the claims and the Specification, we understand that the word “design” in the context of “resetable memory” essentially means a circuit design. Moreover, understanding that paragraph b) in claims 1 and 11 is the implementation step relative to the preceding “inferring” step in paragraph a), this implementation step explains *how* the “resetable memory” is to be used in a semiconductor circuit. Accordingly, the broadest reasonable interpretation of “resetable memory design” in light of the Specification is a resetable memory as defined previously, implemented in a circuit design.

III. ANALYSIS

A. Alleged Anticipation of Claims 1–2, 10–12, and 20 by Vander Zanden

For the reasons given below, despite the arguments provided in the Petition, and the evidence cited therein, Petitioner has not shown, by a preponderance of the evidence, that each of claims 1–2, 10–12, and 20 are unpatentable as anticipated by Vander Zanden.

1. Vander Zanden

Vander Zanden discloses a method of creating descriptions of small memory designs “such as multi-port register files” for use in computer emulation—synthesis—that differs from conventional techniques. Ex. 1003, 71. The method synthesizes a description of a memory, specifically a two-dimensional array *regFile*, apart from other logic and datapath elements as a behavioral description. *Id.* at 72. “This allows the designer to include the behavioral description of the memory with the rest of the HDL [Hardware Description Language] code, yet offers multiple architectural

Case IPR2014-00287

Patent 6,836,420

implementations for the memory without additional effort for the designer.”

Id. at 71. Figure 3 of Vander Zanden is reproduced below:

```
architecture regfilesyn_A of regfilesyn is
begin
  process(clk,rst)
    subtype addrType is integer range 0 to 15;
    type regFileType is array (addrType) of integer;
    variable regFile: regFileType;
  begin
    if rst='1' then
      out1 <= (others => '0');
    elsif clk'event and clk='1' then
      if s2='1' then
        out1 <= regFile(addr1) + regFile(addr2);
      else
        out1 <= regFile(addr3);
      end if;
      if s1='1' then
        regFile(writeAddr) := dataIn;
      end if;
    end if;
  end process;
end regfilesyn_A;
```

Figure 3: VHDL Description

Figure 3 of Vander Zanden illustrates an HDL model for a memory having an if-else statement that includes on one hand a data out, “out1” result of “0,” and on the other hand, a data out result of cell addresses, “addr1, 2, 3.” Ex. 1003, 73.

Figure 4 of Vander Zanden, reproduced below, discloses a particular datapath circuit design based on the HDL description of Figure 3 including two registers, an adder (ADD1), and a multiplexor leading flip-flop DFF 6 to the data out line OUT1. *Id.*

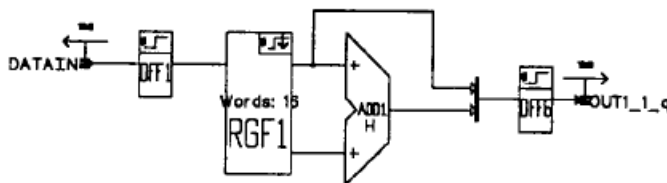


Figure 4: Datapath Design Containing Register File Cell

Case IPR2014-00287

Patent 6,836,420

Figure 4 of Vander Zanden illustrates a datapath design containing register file cell.

According to Vander Zanden, the use of such a memory synthesis technique and datapath design reduces the size of the circuit design as well as operational delay. *Id.* at 74.

2. Discussion

i. Claims 1–2, and 10

Petitioner argues that Vander Zanden anticipates claim 1, a method claim, because it discloses not only “inferring a resetable memory” but also “incorporating a resetable memory design into a design for said semiconductor circuit.” Pet. 15–16. Petitioner states that “[a] POSA would understand the synthesis process described by Vander Zanden to teach inferring the existence of a memory by the presence of a ‘two-dimensional array’ within the behavioral description.” *Id.* at 16 (citing Detjens Decl. ¶ 36(a)). Petitioner asserts that a two-dimensional array of bits, such as *regFile* disclosed by the VHDL model in Vander Zanden’s Figure 3, constitutes a memory, and that resetable flip-flop DFF6 on the data out line as shown in the circuit design at Figure 4 makes the memory resetable. *Id.* (citing Ex. 1003, 72).

Patent Owner argues that Vander Zanden does not disclose “a resetable memory design” as claimed “because, from the perspective of the downstream circuitry, the circuit shown in Vander Zanden Figure 4 is not a memory. This is due to the presence of an adder in the circuit.” PO Resp. 26. In support, Patent Owner cites to Petitioner’s Declarant, Mr. Detjens’, deposition testimony:

Case IPR2014-00287

Patent 6,836,420

Q. Would the average engineer understand [the circuit with an adder] to be a memory?

...

THE WITNESS: The average engineer would not expect an adder to be at that place in a memory.

Id. (citing Ex. 2010, 41:3–16). Patent Owner alleges that Mr. Detjens, as well as Patent Owner’s Declarant, Dr. Hutchings, agree that the downstream circuitry from Vander Zanden’s output flip-flop DFF6 would, at times, see the sum of two memory locations due to adder ADD1, and not an actual memory value from the memory locations in RGF1. *Id.* at 28. Patent Owner contends that because of this a designer of ordinary skill in the art “could not (and would not) use the Vander Zanden Figure 4 circuit as a memory because in one state the circuit would output values [from ADD1] completely different from either a reset value or a value stored in the register file RGF1. *Id.* at 29.

Petitioner’s Declarant, Mr. Detjens testifies that at the time of the filing of the patent, those of ordinary skill in the art would have understood Vander Zanden to disclose a memory, specifically by the steps of:

[a]nalyzing the extracted logic equations to determine the characteristics, i.e., requirements, of the memory described in the behavioral description, e.g., whether the memory is written synchronously or asynchronously and how many read and write ports are required for the memory. (*Id.*, Step 4.) A POSA would have understood another characteristic to be whether the memory is resettable.

Ex. 1002 ¶ 36.e. Mr. Detjens states that one of ordinary skill in the art would consider Figures 3 and 4 in Vander Zanden to disclose a memory, even though there is an adder on one of the outputs of the circuit because the

Case IPR2014-00287

Patent 6,836,420

“ADDER only affects the output of the circuit in one mode of the circuit, when the multiplexer selects the ADDER path.” Ex. 1024 ¶¶ 23–24 (citing Ex. 1003, 74, Fig. 4). In a mode of the circuit where the multiplexer does not select the ADDER, Mr. Detjens alleges that “a POSA would have understood the circuit of Fig. 4 to operate as memory, with the output value of RGF1 appearing on the output, unless the reset signal, ‘rst,’ has been applied.” *Id.* at 24. Thus, Mr. Detjens alleges that on one hand the circuit discloses a memory with a reset function, and on the other hand it discloses a memory output operated upon by ADDER.

Petitioner’s evidence is persuasive that a person of ordinary skill in the art would consider the circuit shown in Vander Zanden’s Figure 4 to operate as a memory, as we have construed the term, with a reset capability. We construed “memory” as a device for storing data having a plurality of cells, each cell having a unique address for storing data, where data is written to a cell during a write function, and, during a read function data is retrieved from a cell and presented at a memory output. *See* section II.B.2. Although the additional circuitry of the ADDER is provided to perform an operation on values output from RGF1, in the mode where the ADDER is not selected, OUT1 will output a value from a register in RGF1, unless a reset signal is applied, as a memory would in accordance with our claim construction. At least in one state downstream circuitry would recognize the circuit of Figure 4 as a memory, thus “inferring” the circuit to be a memory with reset capability. Patent Owner’s argument that this does not occur when the ADDER mode is in effect, does not explain why the portion of the circuit without the ADDER circuitry does not operate and would not be recognized, as a memory having a reset function.

Case IPR2014-00287

Patent 6,836,420

Patent Owner further asserts that Vander Zanden does not anticipate the claimed subject matter because it does not disclose a “resetable memory,” as properly construed, because Vander Zanden does not keep, i.e. remember, that the memory has been reset. PO Resp. 29–33. Patent Owner contends that the code in Vander Zanden’s Figure 3 includes well known “if then” statements which show that only when the reset signal is high, e.g. $\text{rst}=1$, is OUT1 a reset value “0.” Otherwise, when the signal goes low, “the output of the circuit (out1) is assigned the value stored in a particular memory address (see line 14) or the sum of two values stored in different memory addresses (*see* line 12).” PO Resp. 32 (citing Ex. 2011 ¶ 59).

Based on our claim construction, we are persuaded that Vander Zanden does not disclose a “resetable memory” as recited in independent claim 1. We determined that a “resetable memory” means a memory that outputs a reset value until new data is written into the memory. *See* section II.B.3. Vander Zanden’s “if then” statement in the code shown in Figure 3 outputs a reset value “0” only when the reset signal is high, in all other modes it outputs either a value from RGF1 or added values from RGF1. *See* Ex. 1003, Fig. 3; *see also* PO Resp. 32. In other words OUT1 does not keep, or remember, the reset value until new data is written to RGF1. We find Patent Owner’s position persuasive that the resetable flip-flop DFF6 on the output line of the circuit as described in Vander Zanden’s Figure 3, and shown in Figure 4, does not render the registers at RGF1 a “resetable memory” as properly construed because it fails to maintain OUT1 at a reset value until new data is written to a cell.

Petitioner’s argument that no temporal limitation should be read into the claim is unpersuasive because, as discussed above in section II.B.3, the

Case IPR2014-00287

Patent 6,836,420

'420 patent expressly differentiates the “reset” functionality of the described “resetable memory” from known resetable memories where the described, and claimed methodology in each embodiment, “causes the reset value **208** to be provided **302** as the effective memory cell output—until the cell has been written to.” Ex. 1001, 4:56–58.

Vander Zanden does not disclose a “resetable memory design” for similar reasons as set forth above with respect to “resetable memory.” We are persuaded that the resetable flip-flop DFF 6 on the output line as shown in Figure 4 does not disclose a “restable memory design” as called for in the claims because Petitioner does not explain how the circuit design in Figure 4 implements the retention of a reset value until new data is written to the cell, where it only toggles between the reset mode, ADDER mode and mode which returns a value from RGF1.

For the reasons provided above, Petitioner has not established by a preponderance of the evidence, the unpatentability of claim 1 as anticipated by Vander Zanden under 35 U.S.C. § 102(b) because each and every element as set forth in the claim is not found, either expressly or inherently described, in Vander Zanden in as complete detail as is contained in claim 1. *Scripps Clinic & Research Found. v. Genentech, Inc.*, 927 F.2d 1565, 1576 (Fed. Cir. 1991) (“There must be no difference between the claimed invention and the reference disclosure, as viewed by a person of ordinary skill in the field of the invention.”).

Because the remaining dependent claims 2 and 10 depend directly from claim 1, and necessarily include all the limitations of claim 1, these claims also are not anticipated by Vander Zanden.

Case IPR2014-00287

Patent 6,836,420

ii. Claims 11–12, and 20

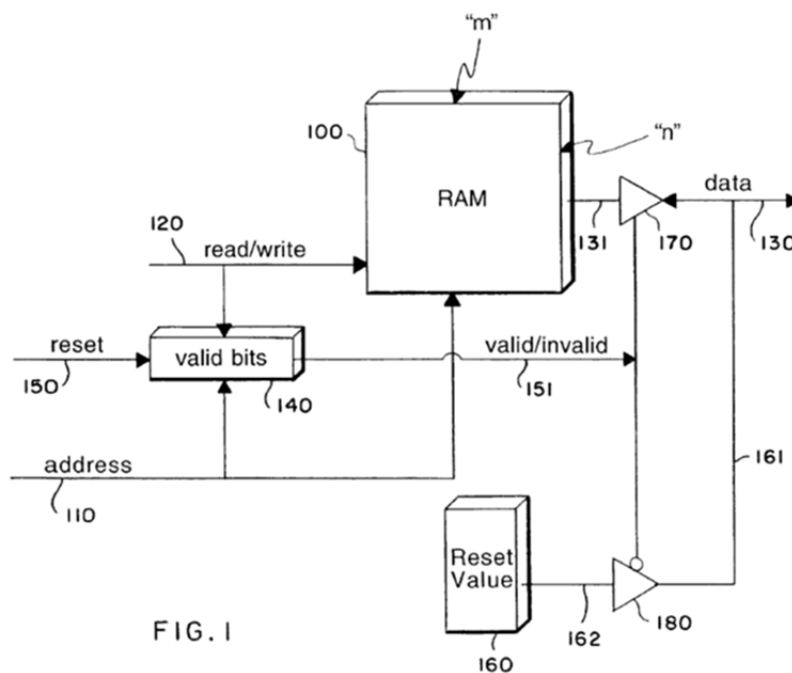
Independent claim 11 is for a “machine readable medium” with instructions to carry out the method with exactly the same limitations in paragraphs a) and b) as discussed above with respect to claim 1. For the same reasons as set forth above, claim 11 and dependent claims 12 and 20 are also not anticipated by Vander Zanden.

B. Alleged Obviousness of Claims 1–3, 10–13, and 20 over Vander Zanden and Shand

For the reasons given below Petitioner has shown, by a preponderance of the evidence, that each of claims 1–3, 10–13, and 20 are unpatentable based on the combination of Vander Zanden and Shand.

1. Overview of Shand

Shand is directed to a method, and specific circuit designs, for resetting memories. Ex. 1006, 1:5–14. Figure 1, reproduced below, illustrates a circuit diagram for a resetable memory according to Shand.



Case IPR2014-00287

Patent 6,836,420

Shand discloses, for example in Figure 1, above, non-resetable RAM 100 and reset register 140 with corresponding bits “m” to respective memory locations in RAM 100. *Id.* at 2:31–66. When a reset register bit value is valid for a particular memory location, “the reset value from register [160] is produced on line 130 via lines 162 and 161, instead of the actual data stored at the addresses memory location.” *Id.* at 2:64-66. In this way, Shand explains, a reset value is maintained until “[a] write to a particular memory location will set the corresponding bit of register 140 to, for example, a logical one, and the corresponding memory location after the write is now valid.” *Id.* at 43–46.

2. Discussion

Patent Owner argues that Shand does not remedy Vander Zanden’s failure to teach the step of “inferring the existence of a resetable memory” as recited in claims 1 and 13, and further, that one of ordinary skill in the art would not combine these references because Shand discloses a structurally and functionally different circuit from Vander Zanden. PO Resp. 53–57.

As discussed previously in section III.A.2.i, with respect to the anticipation challenge to Vander Zanden, we determined that Vander Zanden discloses inferring a “memory” in accordance with our claim construction, but not a “resetable memory.” This is because Vander Zanden’s Figure 4 discloses a memory circuit design including a reset function, however, the evidence does not show that the memory maintains, or remembers the reset value for output, OUT1, until new data is written to RGF1 as required by our claim construction. *See* Ex. 1003, Figs. 3–4.

In the case of obviousness, Petitioner relies upon Vander Zanden’s derivation of a memory circuit from Verilog behavioral description to

Case IPR2014-00287

Patent 6,836,420

disclose the step of inferring the existence of a memory, and turns to Shand for disclosure of a “resetable memory” and a “resetable memory design” for incorporation into a semiconductor circuit. Pet. 31. Petitioner asserts that one of ordinary skill in the art would have combined Shand with Vander Zanden in order to reduce the size of the circuit design. *Id.* at 32 (citing Ex. 1002 ¶ 68).

Patent Owner argues specifically that Petitioner’s Declarant, Mr. Detjens, admitted that one of ordinary skill in the art would not replace the memory in Vander Zanden with the circuit described in Shand. PO Resp. 55. This argument mischaracterizes Mr. Detjens testimony as well as the question. The question asked Mr. Detjens if he could identify “what *memory* in Vander Zanden you would replace with the Shand memory?” Ex. 2010 84:2–3 (emphasis added). Read in context, Mr. Detjens’ initial answer “No” differentiates between replacing the “memory” and replacing the “resetable memory.” *See id.* at 84:1–20. Mr. Detjens continued on, and answered the question with further specificity, stating that “for a *resetable memory* that is created through the process given in the Vander Zanden paper, you would replace it with the memory in the Shand patent.” *Id.* at 84:18–20 (emphasis added). Thus, we do not find Patent Owner’s argument on this point to be persuasive.

Patent Owner further asserts that Shand’s resetable memory is functionally different and one of ordinary skill in the art would not utilize Shand’s resetable memory with Vander Zanden because it “would render the circuit inoperable (*i.e.*, the synthesized circuit in Vander Zanden Figure 4 would no longer match the HDL and it would not work as the designer intended).” PO Resp. 57 (citing *McGinley v. Franklin Sports, Inc.*, 262 F.3d

Case IPR2014-00287

Patent 6,836,420

1339, 1354 (Fed. Cir. 2001)). Petitioner replies that “A POSA knew how to describe a resetable memory in behavioral or RTL level terms, and how to recognize a resetable memory in such a description.” Reply 2–3 (citing Ex. 1024 ¶¶ 10–15; Ex. 1020 at 17–18, 27–28, 75–77, 84–85, 91–98).

Petitioner’s position, and Dr. Detjens’ testimony, are persuasive. Neither Patent Owner’s argument that Shand fails to disclose the inferring step, nor the testimony provided in Dr. Hutchings’s Declaration alleging that Shand discloses only resetable memory circuits and not *inferring* a memory with reset, explains why it was not within the ordinary skill of one in the art to have modified the respective underlying HDL code, for example as shown in Vander Zanden’s Figure 3, to include code defining the respective resetable memory circuit such as disclosed in Shand. *See* PO Resp. 57, *see also* Ex. 2011 ¶¶ 85–88. As discussed above with respect to anticipation, we are persuaded that Vander Zanden discloses the “inferring” step for a memory as recited in independent claims 1 and 13, and that it would have been within the abilities of a person of ordinary skill in the art to combine Shand’s resetable memory circuit design with Vander Zanden and consequently modify the HDL code to address the functional changes imparted by the implementation of a different resetable memory in such a combination.

Patent Owner does not address the specific limitations of dependent claims 2, 3, 10, 12, 13, and 20 but relies, for each of claims 1–3, 10–13 and 20, upon its erroneous analysis with respect to the combination of Vander Zanden and Shand. *See* Prelim Resp. 47–49, *see also* PO Resp. 52–57. We therefore conclude that a preponderance of the evidence demonstrates that

Case IPR2014-00287

Patent 6,836,420

claims 1–3, 10–13 and 20 are unpatentable based on the combination of Vander Zanden and Shand.

C. Alleged Obviousness of Claims 1–3, 10–13, and 20 over Vander Zanden and Runaldue

1. Overview of Runaldue

Runaldue discloses a memory circuit with a zero detect logic to determine if a row of a memory array is to be treated as reset to a zero state, or preserved in a non-zero state according to corresponding tag-memory cells. Ex. 1007, 2:1–16. Runaldue explains that, “[t]he tag-memory cells are resetable to a zero state to indicate that the associated row of the memory array is to be treated as all being in the zero state. This is to be done even though the actual contents of the memory are not all zeros.” *Id.* at 2:11–16.

2. Discussion

Patent Owner states that Petitioner’s obviousness arguments for the combination of Vander Zanden and Runaldue are based on the same “incorrect premise” as Vander Zanden and Shand and “are largely identical to those made with respect to Shand . . . and fail for the same reasons described above.” PO Resp. 58. Patent Owner’s argument that Vander Zanden does not infer a resetable memory is, as discussed above with respect to Shand, incorrect. However, because we construed the term “resetable memory” and “resetable memory design” differently than Petitioner proposed, the question remains whether Runaldue meets both these further claim limitations of a “resetable memory” and a “resetable memory design” as properly construed.

Runaldue discloses in the Background of the Invention, that a “global reset” function is understood as “a condition in which every memory

Case IPR2014-00287

Patent 6,836,420

element is reset to a predefined logical zero state.” Ex. 1007, 1:19–21. In the Summary of the Invention, Runaldue explains that in a memory array, each row of memory cells will have a “tag-memory cell[]” that indicates whether the row has a non-zero, or a zero, state. *Id.* at 2:1–7. Runaldue further discloses that “[t]he tag-memory cells are resetable to a zero state to indicate that the associated row of the memory array is to be treated as all being in the zero state. This is to be done even though the actual contents of the memory are not all zeros.” *Id.* at 11–16. What we cannot determine from our review of Runaldue is that the tag-memory maintains a zero state until new data is written to the memory. Runaldue is silent as to such a requirement. We construed “resetable memory” as a memory that outputs a reset value until new data is written into the memory. Runaldue discloses a memory that outputs a reset value, for example a zero state, but there is no disclosure that the zero state is maintained, or “remembered” until new data is written to the memory. Therefore, even if combined with Vander Zanden, Runaldue fails to correct the underlying deficiency of Vander Zanden. Accordingly, Petitioner has failed to show by a preponderance of the evidence that the combination of Vander Zanden and Runaldue meets all the limitations of independent claims 1 and 11. Because the remaining dependent claims 2, 3, 10, 12, 13 and 20 depend directly from claim 1 or claim 11, and necessarily include all the limitations of their respective dependent claims, these claims also are not rendered obvious by Vander Zanden and Runaldue.

D. Alleged Anticipation of Claims 1–2, 10–12, and 20 by Wohl

For the reasons given below, despite the arguments provided in the Petition, and the evidence cited therein, Petitioner has not shown, by a

Case IPR2014-00287

Patent 6,836,420

preponderance of the evidence, that each of claims 1–2, 10–12, and 20 are unpatentable as anticipated by Wohl.

1. Overview of Wohl

Wohl describes an improvement in integrated circuit design flow that more efficiently uses a single Verilog simulation library for both simulation and ATPG (“Automatic Test Pattern Generator”) model building. Ex. 1004, 1011. Wohl explains that using a single library in the netlist reader and model builder of an ATPG tool eliminates the necessity for coding and verifying a separate ATPG library, and additionally simplifies debugging test problems. *Id.*

Wohl specifically discloses an ATPG model builder for converting a behavioral Verilog description, for example of a RAM including a reset line and read/write ports, into a resulting ATPG model test pattern generator. *Id.* at 1017, Figs. 6–7. Wohl discusses a methodology which rewrites, or recodes, the Verilog RAM description into a simplified form that is sufficiently simple for automatic processing and displays the resulting ATPG RAM model in a schematic viewer. *Id.* By way of example, the RAM ATPG model shown in Wohl’s Figure 7 below, appears as a test pattern generator simulated by the behavioral Verilog description shown in Figure 6.

Figure 6 of Wohl, reproduced below, is a RAM described in behavioral Verilog having a reset line. Ex. 1004, 1017.

Case IPR2014-00287

Patent 6,836,420

```

module withram(reset, r, w, a, d1, d2);
input reset, r, w;
input [3:0] a;
input [7:0] d1;
output [7:0] d2;
reg [7:0] mymem [15:0], d2;
integer i;
always @ reset if (reset)
for (i=0; i<16; i=i+1) mymem[i] <= 0;
always @ (posedge w) mymem[a] <= d1;
always @ r if (r) d2 <= mymem[a];
else d2 <= 0; // read_off is 0
endmodule

```

Figure 6. Simple RAM with reset, write and read port.

Wohl's Verilog description, shown above in Figure 6, results in the "unambiguous ATPG model of Figure 7," displayed in the block diagram reproduced below.

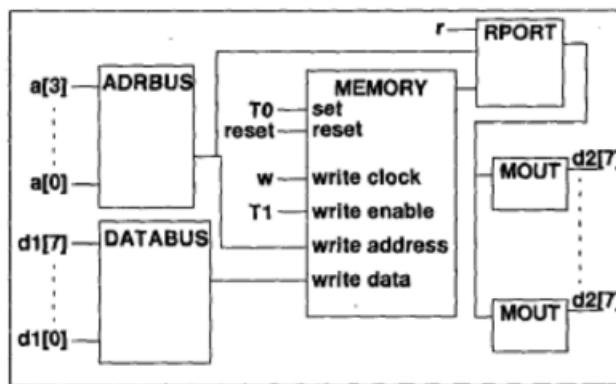


Figure 7. ATPG model of the RAM in Figure 6.

Wohl's Figure 7, illustrated above, is described as "the ATPG model built as displayed by the schematic viewer." *Id.*

2. Discussion

i. Claims 1–2 and 10

Petitioner argues that Wohl discloses each limitation of claim 1 including a method for converting a behavioral Verilog description of RAM with resettable memory into a gate-level description of an ATPG model of a circuit design including resettable memory. Pet. 19–20 (citing Ex. 1004,

Case IPR2014-00287

Patent 6,836,420

1017, Figs 6–7). Petitioner specifically asserts that because Wohl teaches a behavioral description of a RAM with a reset line, and an ATPG model builder that converts the behavioral description into an ATPG model “[a] POSA would understand this conversion to require inferring, or recognizing, the existence of a resettable memory from the behavioral Verilog description” as called for in paragraph a) of claim 1. Pet. 20 (citing Ex. 1002 ¶¶ 46–47). Also Petitioner contends, Wohl’s ATPG model in Figure 7 would be understood by a person of ordinary skill in the art as a design for a semiconductor circuit having a resettable memory as recited in paragraph b) of claim 1. *Id.* at 20–21 (citing Ex. 1002 ¶ 48).

Patent Owner makes three arguments as to why Wohl does not anticipate the challenged claims. PO Resp. 40. First, Patent Owner argues that Wohl does not disclose a “resettable memory design” as claimed because Wohl discloses only a memory with intrinsic reset, i.e. a resettable memory. *Id.* at 40–43. Second, Patent Owner argues that the ATPG model in Figure 7 does not match the functionality shown in the code in Figure 6, and third, that Wohl does not disclose a semiconductor design. *Id.* at 44–49.

With respect to whether Wohl’s ATPG model in Figure 7 discloses a “resettable memory design” in accordance with the proper claim construction, we determine that it does not. Wohl explains that the ATPG model is part of the development and “typical design flow of integrated circuits,” but provides no explanation of how to derive a memory circuit design, that could be incorporated into a semiconductor circuit, from the design flow. Ex. 1004, 1011. Clearly, an ATPG test model is a schematic representation of the behavioral Verilog description, however, it is not clear from Wohl that one of ordinary skill in the art would understand the ATPG

Case IPR2014-00287

Patent 6,836,420

model to be representative of a circuit design implementing, for example, the memory function described by the behavioral Verilog in Wohl.

Petitioner states in its Reply that the '420 patent does not limit “‘semiconductor circuit’ designs to final designs that can be used, without alteration or optimization, to directly manufacture semiconductor circuits in silicon chips.” Reply 10. Petitioner argues, essentially, that a gate-level ATPG model, is sufficient to disclose a memory circuit design as claimed. *Id.* This argument misses the mark however, because it is unsupported by reliance on any credible evidence, and does not explain where, or how, Wohl’s gate level ATPG model discloses a “resetable memory circuit design,” as properly construed, that can be incorporated into a semiconductor circuit. *See id.* To establish anticipation, every element and limitation of the claimed invention must be found in a single prior art reference, arranged as in the claim. *Karsten Mfg. Corp. v. Cleveland Golf Co.*, 242 F.3d 1376, 1383 (Fed. Cir. 2001).

Petitioner relies upon Mr. Detjens’ testimony alleging that a person of ordinary skill in the art would understand that Wohl teaches “incorporating a resetable memory design into a semiconductor circuit design.” Pet. 21 (citing Ex. 1002 ¶ 48.) Mr. Detjens alleges that “the inference of the memory described in Figure 6’s behavioral Verilog leads to the incorporation of a resettable memory into a gate-level ATPG model description of the circuit, as clearly illustrated in Figure 7.” Ex. 1002 ¶ 48. Mr. Detjens’ Declaration, however, is similarly deficient with respect to any explanation or opinion regarding the understanding of a person of ordinary skill as to how or why the memory block shown in Wohl’s Figure 7 is representative of a circuit. *Id.* The Declaration concludes that “A POSA

Case IPR2014-00287

Patent 6,836,420

would have understood Figure 6 of Wohl as describing the operational flow of a particular circuit” without explaining how the operational flow shown in Figure 7 and code statements from Figures 6 details any circuit structure that could be incorporated into a semiconductor circuit. *Id.* ¶ 49. Indeed, when asked during his deposition to explain what it means to be an ATPG model of a memory, Mr. Detjens testified that “[t]he ATPG model is used to create the test vectors that are part of the design of the semiconductor device.” Ex. 2010, 64. Mr. Detjens’ testimony indicates that test vectors are a functional description necessary in the circuit design process, but, when questioned on this point, Mr. Detjens did not specify that the ATPG model in Figure 7 alone was sufficient to indicate to one of skill in the art a particular circuit or circuit design structure:

A: When you walk up to a foundry and say make this chip, you don’t hand them just a netlist of library elements and say manufacture this. You have to hand them a set of test vectors at the same time that corresponds to that. If you don’t, they won’t manufacture it for you.

Q: So are you telling me that the test vectors, in part, define the circuitry that goes onto the chip?

A: It defines the functionality of the circuitry on the chip.

Id.

We are not persuaded that Petitioner has shown by a preponderance of the evidence that the ATPG test model, as a representation of a “resetable memory,” without more, would be understood by one of ordinary skill in the art to disclose a “resetable memory circuit design” as properly construed, for incorporation into a semiconductor circuit, such that claim 1 is anticipated by Wohl under 35 U.S.C. § 102(b).

Case IPR2014-00287

Patent 6,836,420

Because the remaining dependent claims 2 and 10 depend directly from claim 1, and necessarily include all the limitations of claim 1, these claims also are not anticipated by Wohl.

ii. Claims 11, 12, and 20

Independent claim 11 contains exactly the same limitations in paragraphs a) and b) as discussed above with respect to claim 1. For the same reasons as set forth above, claim 11 and dependent claims 12 and 20 are also not anticipated by Wohl.

E. Motion to Exclude

After institution of trial, Patent Owner filed a Patent Owner Response (Paper 21), along with the Declaration of Dr. Hutchings (Ex. 2011). Petitioner then filed a Petitioner's Reply (Paper 22) as well as a Reply Declaration by Mr. Detjens (Ex. 1024). Subsequently, Patent Owner filed a Motion to Exclude Evidence ("Mot. to Exclude") (Paper 25) arguing, among other things, that the Board should exclude, in their entirety, Petitioner's Reply and Mr. Detjens' Reply Declaration.

1. Petitioner's Reply and Mr. Detjens' Reply Declaration (Ex. 1024)

Patent Owner contends that Petitioner's Reply attempts to advance a new theory that Vander Zanden's output flip-flop DFF6, "*is itself* the resetable memory" along with a corresponding new claim construction that would read flip-flop DFF6 as a "resetable memory." Mot. to Exclude 5. We do not understand Petitioner's Reply as a new theory or claim construction because Petitioner's Reply essentially reiterated their initial proposed construction that a "resetable memory" should include a "memory unit comprising one or more storage cells." *Compare* Pet. 11, *and* Reply 5. In

Case IPR2014-00287

Patent 6,836,420

any event, we determined that on this record, and in accordance with the Specification of the '420 patent, that “resetable memory” should be construed as a memory “that outputs a reset value until new data is written into the memory.” *See* Section II.B.3. Indeed, our claim construction is essentially that urged by Patent Owner in this proceeding.

Patent Owner also complains that Petitioner raised a new single-reference obviousness theory, relying on Vander Zanden, in the Reply. Mot. to Exclude 6. We understand the Petition, and the Reply, to rely upon Vander Zanden to disclose the “inferring” step recited in each of the independent claims 1 and 11. As discussed above in section III.A.2.i, denying anticipation over Vander Zanden, we found that Vander Zanden discloses the step of “inferring” the existence of a memory with reset capability, but not a “resetable memory,” as would be understood by one of ordinary skill in the art in the context of the '420 patent. To the extent Petitioner alleges that the “inferring” step was obvious in view of Vander Zanden in the Reply, we determined that Vander Zanden disclosed this step in our anticipation analysis, and that the elements missing from Vander Zanden, are properly found in Shand, as discussed above in section III.C.2.

Patent Owner next argues that Petitioner presents a “new position” in the Reply that “Vander Zanden’s tool expressly synthesized writes and control signals . . . and the conditions under which each write operation will take place.” Mot. to Exclude 7 (*citing* Reply 3). We do not understand this to be a new position or theory, but a description of memory functions in general. In the context of Vander Zanden’s disclosure of a memory Petitioner’s description is consistent with our claim construction for “memory” as set forth above in Section II.B.2. Petitioner’s reliance in the

Case IPR2014-00287

Patent 6,836,420

Reply upon Figure 8 in Wohl is also not a new theory as Patent Owner asserts because Patent Owner raised the timing issue in its Response based on the parallel executed always statements in Wohl's Figure 6 as being "fatal" to anticipation. *See* PO Resp. 44. Petitioner's Reply merely points out that Figure 8 in Wohl, and the related description, is a further example of behavioral Verilog code that expressly addresses, and corrects, the timing issue from Figure 6. This response, to an issue raised by Patent Owner, that does not go outside the evidence and prior art of record, does not rise to the level of a new argument.

We have reviewed Mr. Detjens' Declaration in Support of Petitioner's Reply, (Ex. 1024) and find the Declaration, as a whole, complies with our rule, 37 C.F.R. § 42.23(b), that a reply may only respond to arguments raised in patent owner's response. For example, Mr. Detjens explains in the Reply Declaration that his testimony is specifically in response to Patent Owner's argument that Vander Zanden does not disclose inferring resettable memories. Ex. 1024 ¶ 5 (citing PO Resp. 29–30). Within the context of Vander Zanden and his previous testimony, Mr. Detjens' Reply Declaration provides appropriate elucidating evidence with respect to the level of ordinary skill in the art in addition to that in his original Declaration. *Compare id.* ¶¶ 6–11, with Ex. 1002 ¶¶ 35–43. The Reply Declaration does, in several paragraphs, refer to certain evidence such as Exhibits 1022 and 1023 not filed with the Petition, but overall, relies substantially upon the prior art and evidence submitted with the Petition. Patent Owner's Motion does not apprise us that the Reply or Reply Declaration is raising a new issue or injecting improper new evidence into the proceeding. Our review of the Reply Declaration indicates that overall Mr. Detjens' testimony is proper

Case IPR2014-00287

Patent 6,836,420

rebuttal evidence, timely submitted, in reply to Patent Owner's Response. Therefore, the request to exclude Petitioner's Reply and Mr. Detjens' Reply Declaration (Ex. 1024) is denied.

2. Exhibits 1005, 1017–1019, 1021–1023, and 1025–1034

Patent Owner moves to exclude Exhibits 1005, 1017, 1018, and 1019 as irrelevant. Mot. to Exclude 2–3. Patent Owner moves to exclude Exhibits 1022 and 1023 which embody material it argues Petitioner should have addressed in the Petition. *Id.* at 8–10. Patent Owner further moves to exclude Exhibits 1021–1023 and 1025–1034 as irrelevant or related to other exhibits Patent Owner asserts should be excluded. *Id.* at 12–13. Patent Owner moves also to exclude Exhibit 1022 under F.R.E. 901, and Exhibits 1022, 1026–1030, and 1032 as hearsay. *Id.* at 14–15. Exhibits 1026–1032 and 1034 are not of record in this proceeding. We do not rely on any of the remaining exhibits 1005, 1017–1019, 1021–1023, 1025 and 1033. Accordingly, Patent Owner's Motion is dismissed as moot as to these exhibits.

IV. CONCLUSION

We conclude that Petitioner has demonstrated by a preponderance of the evidence that (1) claims 1–3, 10–13, and 20 of the '420 patent are unpatentable as obvious over the combination of Vander Zanden and Shand.

This is a final written decision of the Board under 35 U.S.C. § 318(a). Parties to the proceeding seeking judicial review of this decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

V. ORDER

For the reasons given, it is

Case IPR2014-00287

Patent 6,836,420

ORDERED that claims 1–3, 10–13, and 20 of U.S. Patent No. 6,836,420 B1 are determined by a preponderance of the evidence to be unpatentable;

FURTHER ORDERED Patent Owner’s Motion to exclude Exhibits 1021–1034 is denied-in-part and dismissed-in-part;

FURTHER ORDERED that because this is a final written decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

Case IPR2014-00287

Patent 6,836,420

PETITIONER:

Andrew M. Mason
John D. Vandenberg
Jeffrey S. Love
KLARQUIST SPARKMAN, LLP
andrew.mason@klarquist.com
john.vandenberg@klarquist.com
Jeffrey.love@klarquist.com

PATENT OWNER:

William H. Wright
Travis Jensen
ORRICK, HERRINGTON, &
SUTCLIFFE LLP
wwright@orrick.com
tjensen@orrick.com

U.S. Patent No. 6,836,420

(12) **United States Patent**
Seshadri et al.

(10) **Patent No.:** **US 6,836,420 B1**
(45) **Date of Patent:** **Dec. 28, 2004**

(54) **METHOD AND APPARATUS FOR
RESETABLE MEMORY AND DESIGN
APPROACH FOR SAME**

(75) Inventors: **Vijay K. Seshadri**, San Jose, CA (US);
Kenneth S. McElvain, Los Altos, CA
(US)

(73) Assignee: **Synplicity, Inc.**, Sunnyvale, CA (US)

(*) Notice: Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 265 days.

(21) Appl. No.: **10/091,787**

(22) Filed: **Mar. 4, 2002**

(51) Int. Cl.⁷ **G06F 17/50**

(52) U.S. Cl. **365/51**; 365/63; 716/18;
716/2; 716/3

(58) Field of Search 365/51, 63; 716/18,
716/2, 3

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,831,868 A * 11/1998 Beausang et al. 716/18
5,903,466 A * 5/1999 Beausang et al. 716/18
6,289,498 B1 * 9/2001 Dupenloup 716/18

6,292,931 B1 * 9/2001 Dupenloup 716/18
6,295,636 B1 * 9/2001 Dupenloup 716/18
6,298,472 B1 * 10/2001 Phillips et al. 716/18
6,378,123 B1 * 4/2002 Dupenloup 716/18
6,421,818 B1 * 7/2002 Dupenloup et al. 716/18
6,463,560 B1 * 10/2002 Bhawmik et al. 714/733
6,539,523 B1 * 3/2003 Narain et al. 716/5
6,598,209 B1 * 7/2003 Sokolov 716/4
6,609,229 B1 * 8/2003 Ly et al. 716/4

FOREIGN PATENT DOCUMENTS

JP 408153126 A * 6/1996 G06F/17/50

* cited by examiner

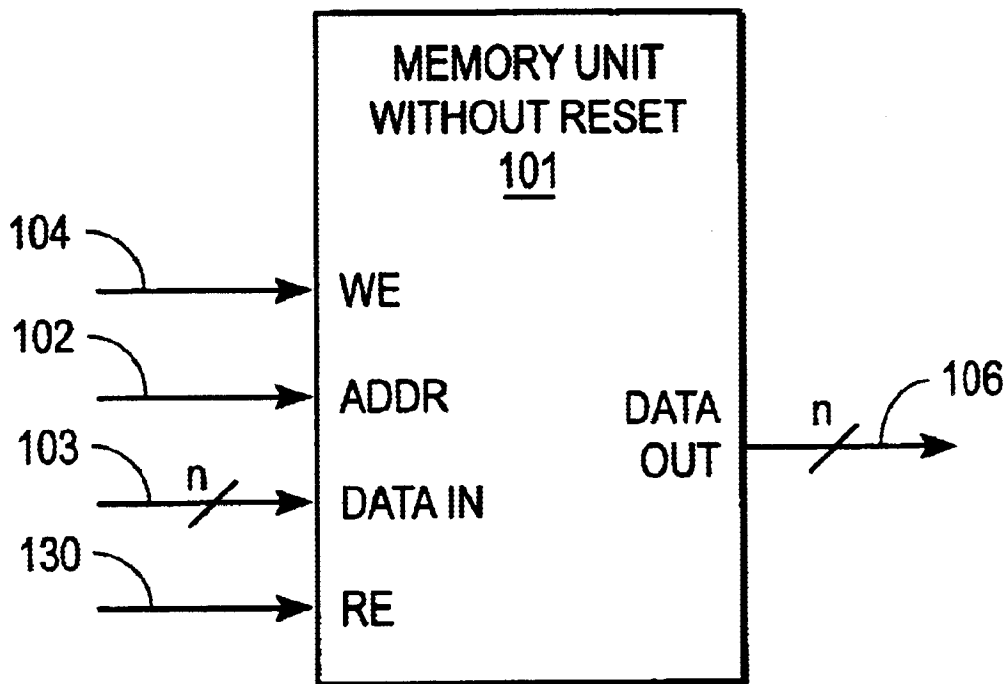
Primary Examiner—Viet Q. Nguyen

(74) *Attorney, Agent, or Firm*—Blakely, Sokoloff, Taylor &
Zafman LLP

(57) **ABSTRACT**

A resetable memory is described that includes a memory
without reset capability having a data output coupled to a
first input of a first multiplexer. A second input of the first
multiplexer has a reset value input. A channel select for the
first multiplexer is coupled to a resetable storage cell output
that indicates whether a storage cell within the memory
without reset capability has been written to after a reset or
has not been written to after a reset.

20 Claims, 10 Drawing Sheets



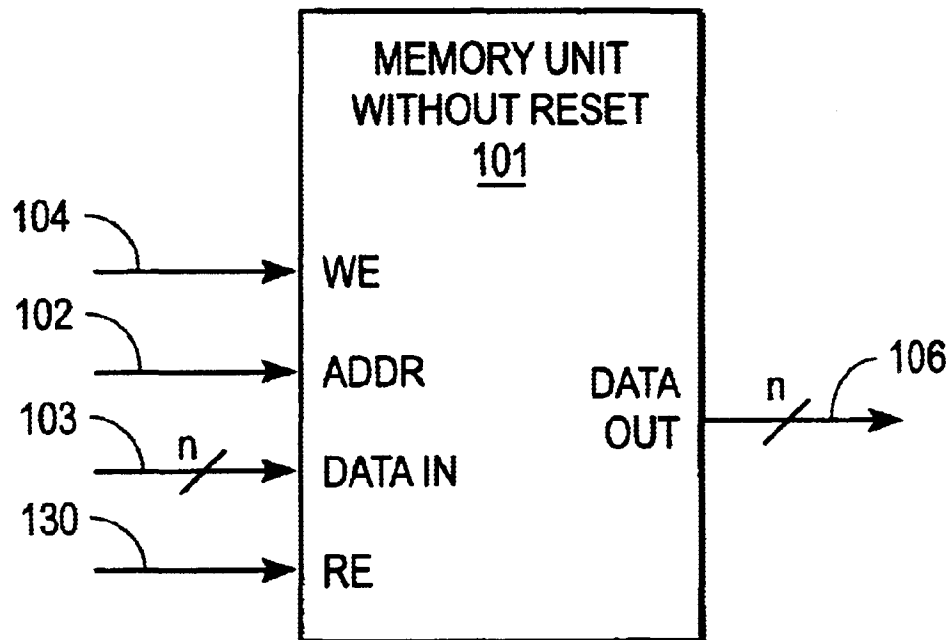


FIG. 1

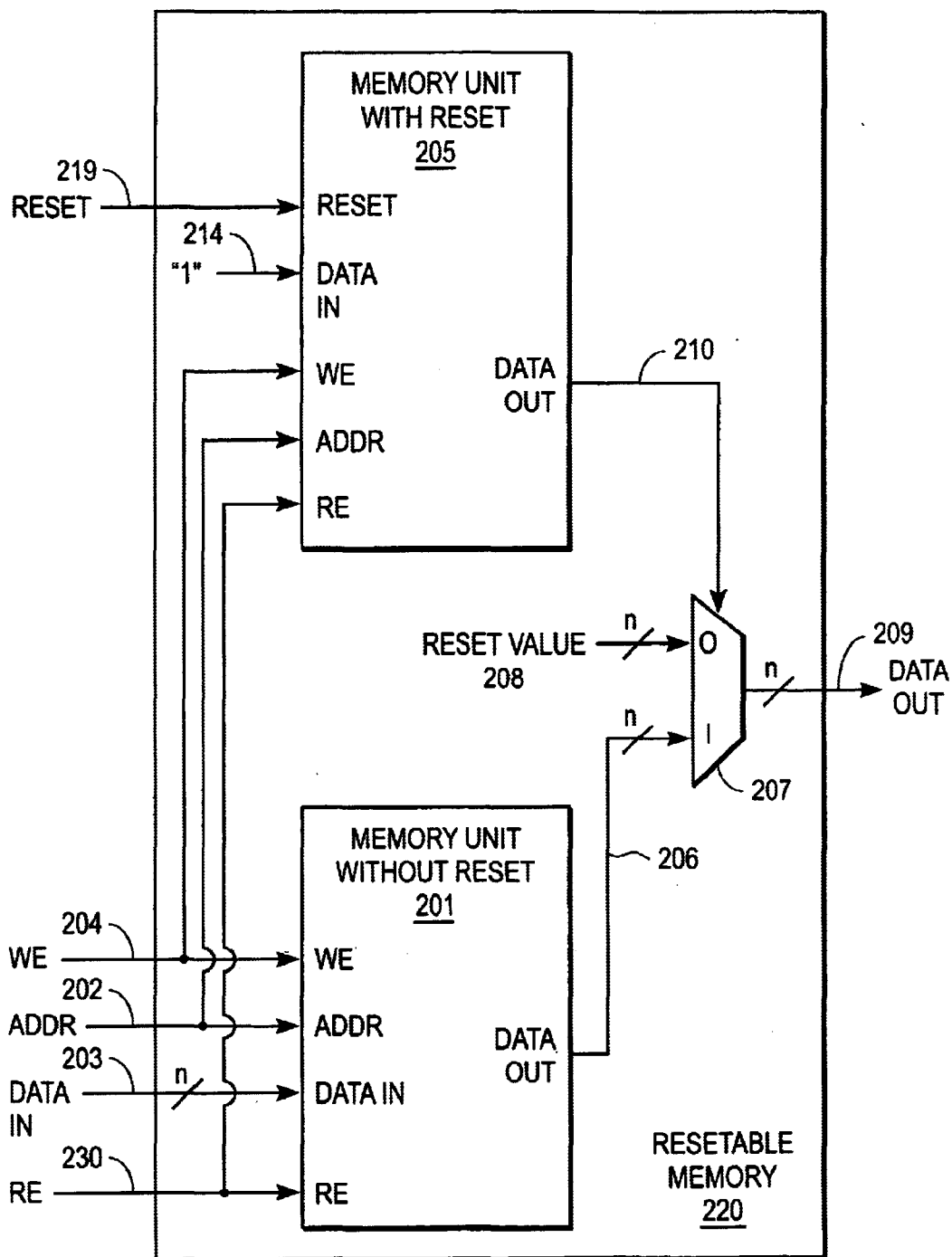


FIG. 2A

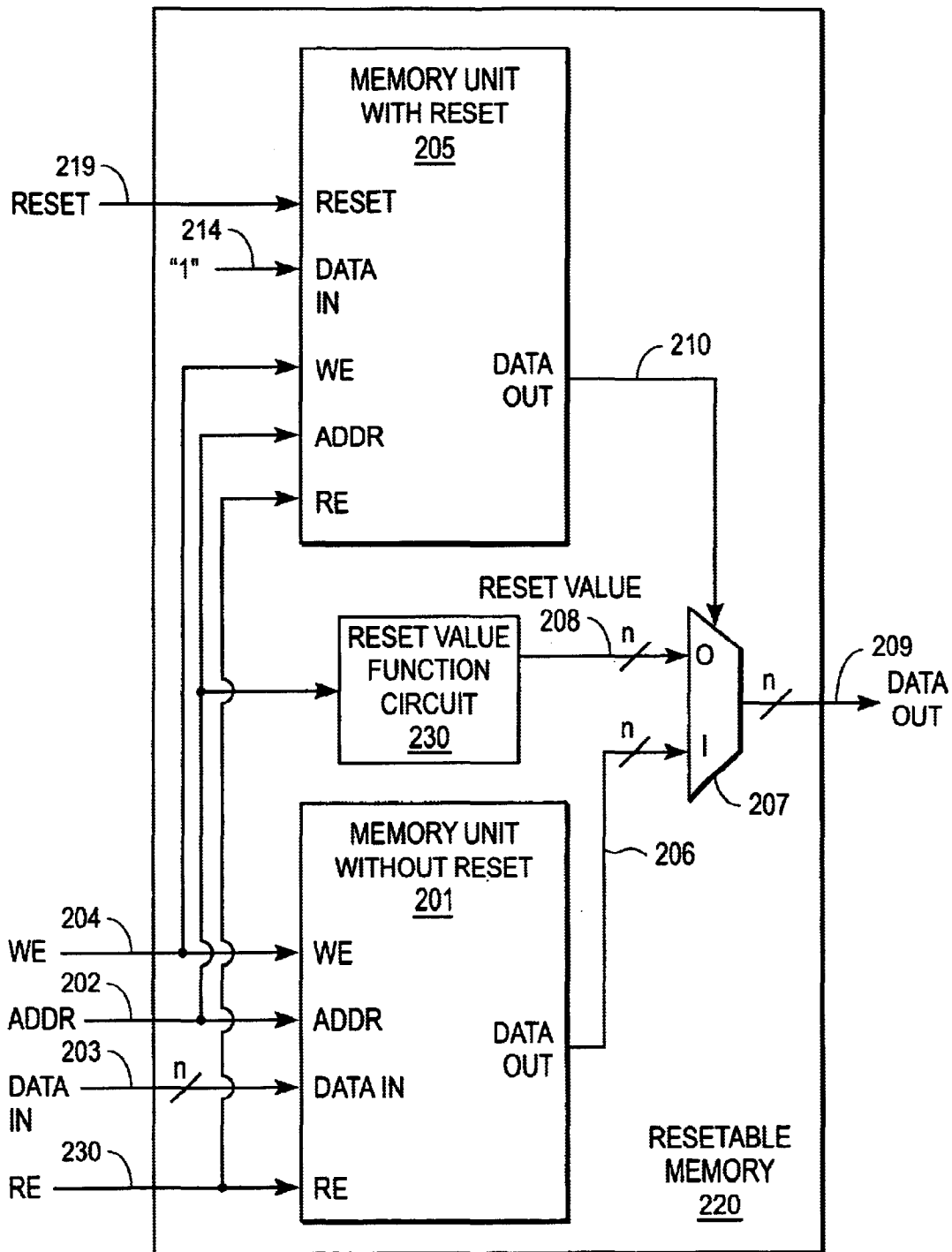


FIG. 2B

U.S. Patent

Dec. 28, 2004

Sheet 4 of 10

US 6,836,420 B1

```

module synReset(data_in, addr, reset, we, clk, data_out);

parameter data_width = 1024;
parameter addr_width = 10;
parameter RAMsize = 8;
parameter reset_value = 8'D0;

input [data_width-1:0] data_in;
input [addr_width-1:0] addr;
input reset, we, clk;
output [data_width-1:0] data_out;

integer i;
reg [data_width-1:0] mem [RAMsize-1:0];
wire [data_width-1:0] data_out;
//synthesis loop_limit 2000
always @(posedge clk)
begin
    if(reset == 1'b1)
    begin
        for (i=0; i < RAMsize ; i=i+1)
        begin
            mem[i] = reset_value;
        end
    end else if(we == 1'b1)
    begin
        mem[addr] = data_in;
    end
end
assign data_out = mem[addr];
endmodule

```

FIG. 2C

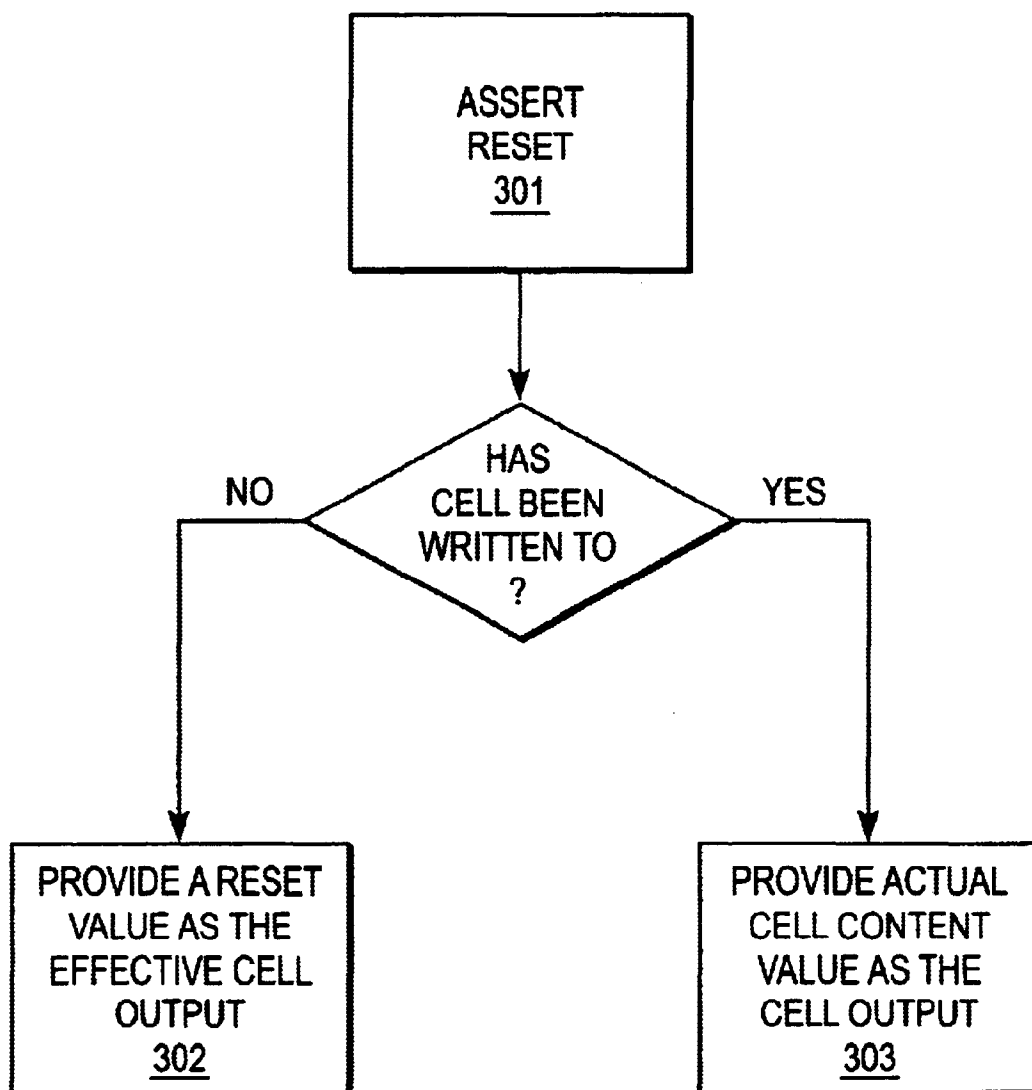


FIG. 3

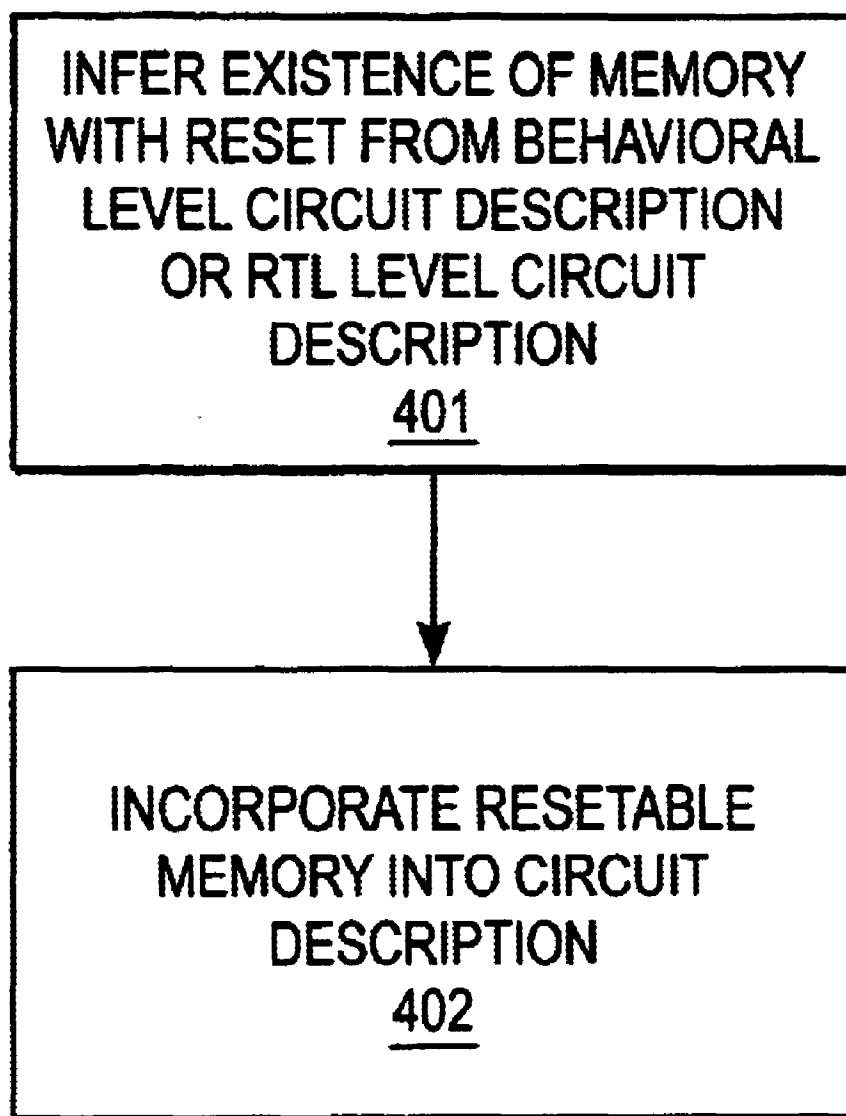


FIG. 4

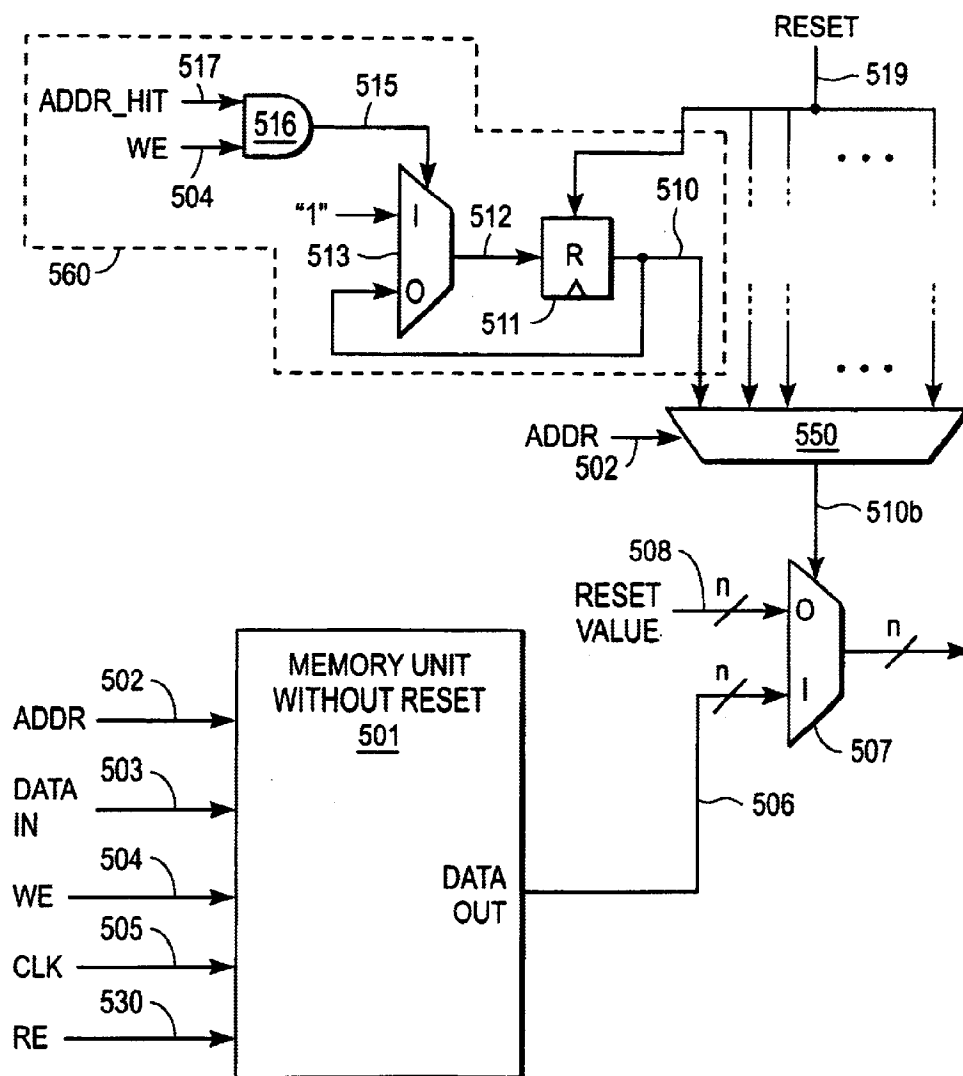


FIG. 5

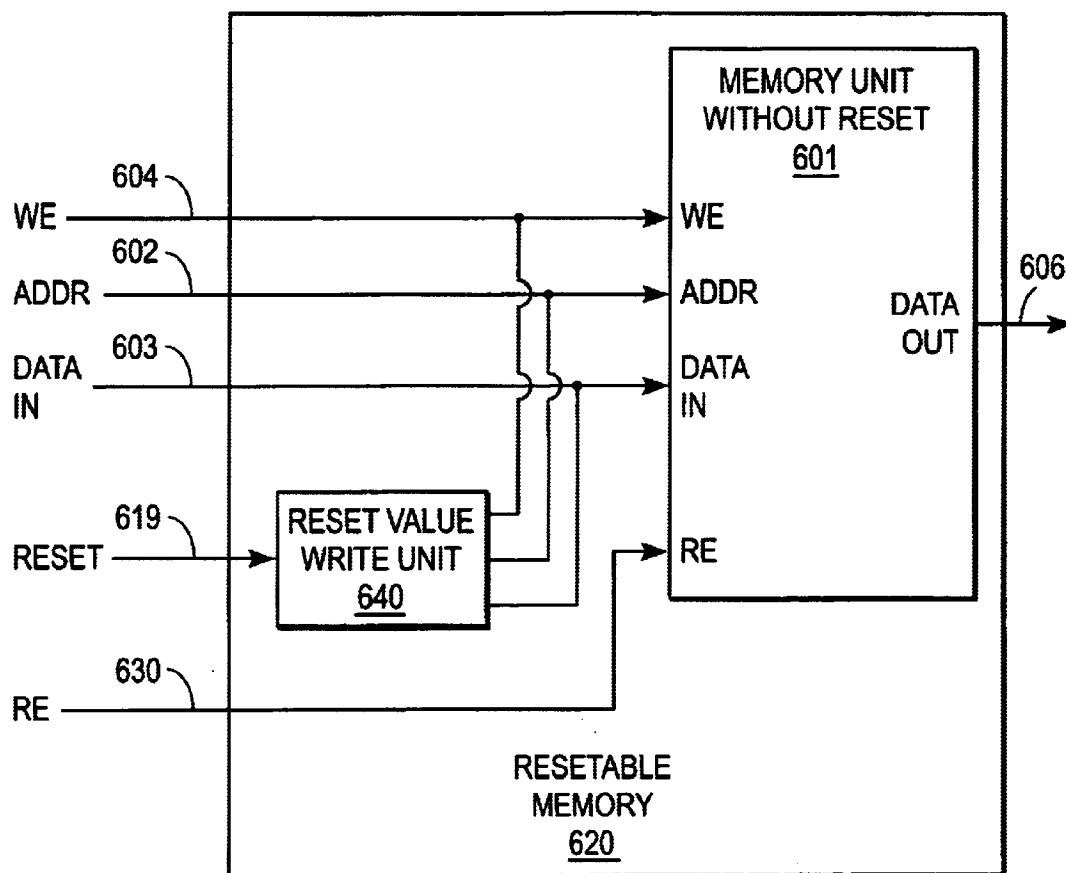


FIG. 6

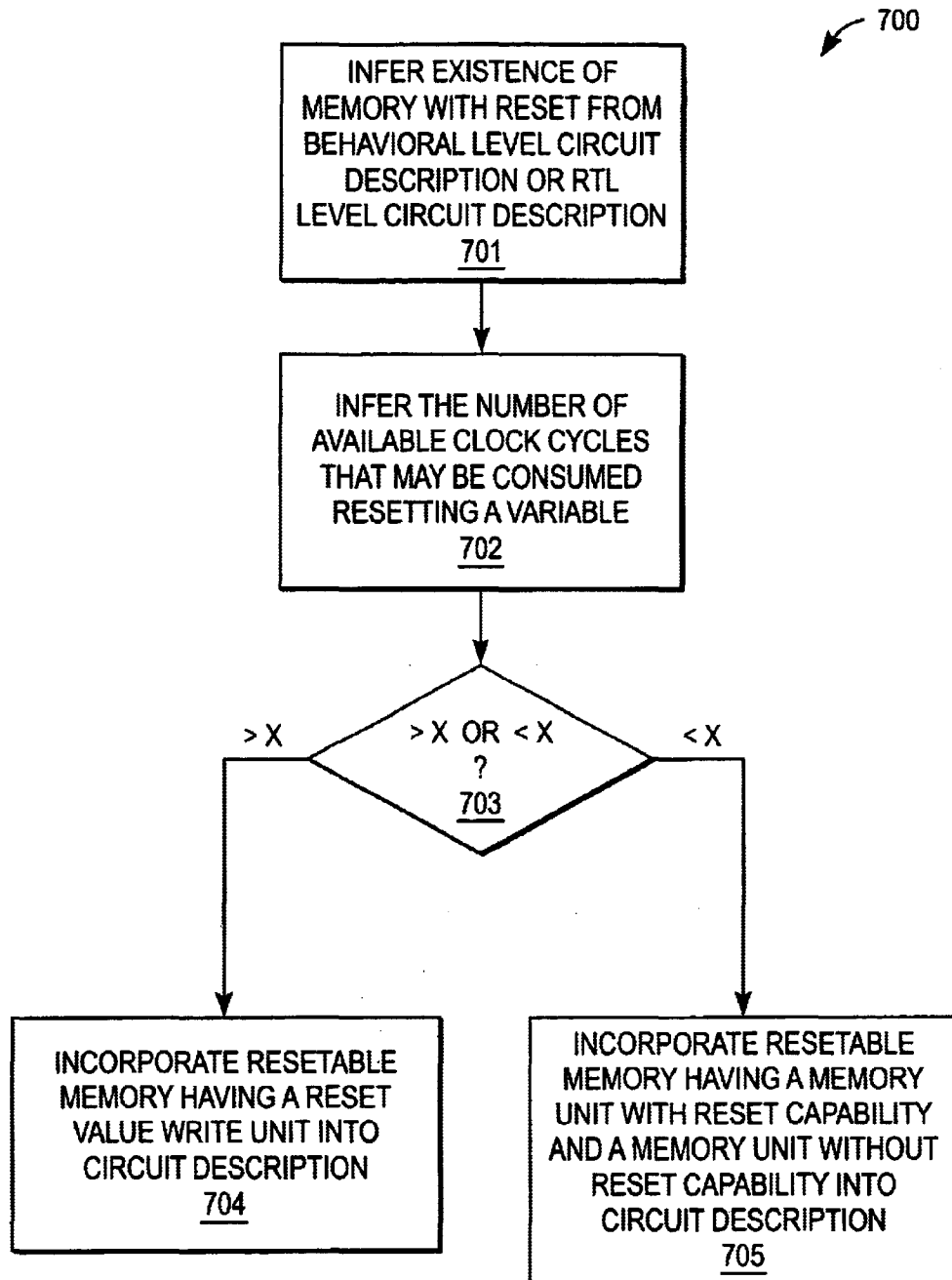


FIG. 7

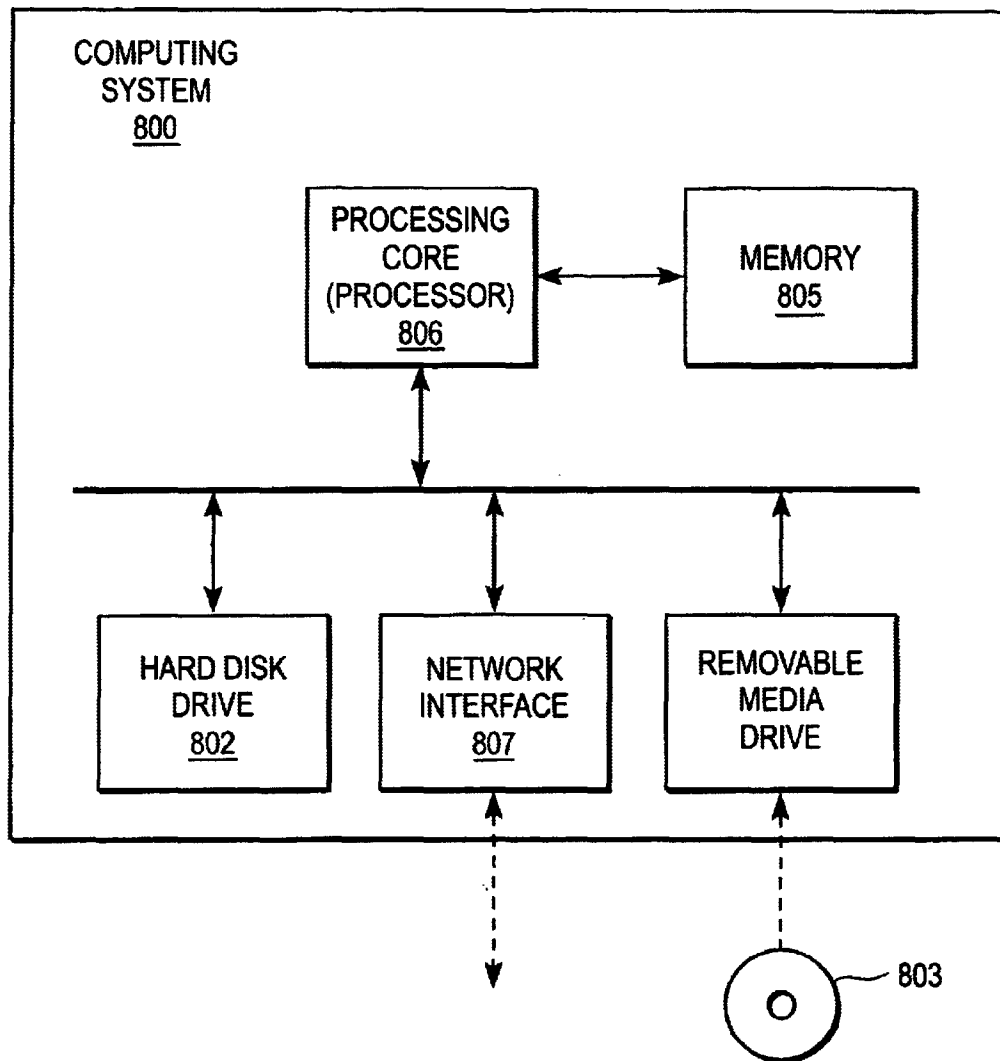


FIG. 8

US 6,836,420 B1

1

METHOD AND APPARATUS FOR RESETABLE MEMORY AND DESIGN APPROACH FOR SAME

FIELD OF THE INVENTION

The field of invention relates generally to electronic circuit design; and more specifically, to a method and apparatus for a resetable memory and a design approach for same.

BACKGROUND

FIG. 1 shows a memory unit **101**. The memory unit **101** may be viewed as having a plurality of storage cells (or simply, "cells"). Associated with each cell is a unique address that provides access to the location of a particular storage cell. Each storage cell has the capacity to store "n" bits (where n is an integer greater than or equal to one). The n bits may be collectively referred to as a word of data.

Often, a memory unit **101** is written to by: 1) providing a word of data (i.e., "n" bits of data as seen in FIG. 1) to be written into the memory unit **101** on a data bus (such as input data bus **103** observed in FIG. 1); 2) providing an address (e.g., on address bus **102**) that defines which storage cell will store the word of data; and 3) presenting a signal to the memory unit that effectively indicates the word of data on the data bus is to be written into the memory unit (such as activating the write enable (WE) control line **104** of FIG. 1). Alternatively, the write could be time based as in a shift register, or it could be a mix of address and time based write.

Often, a memory unit **101** is read from by: 1) providing an address (e.g., on address bus **102**) that defines which cell a word of data will be read from; and 2) presenting a signal to the memory unit that effectively indicates a word of data is to be read from the memory unit (such as activating the read enable (RE) control line **130** of FIG. 1). The word of data is presented at the data output bus **106**. In embodiments alternative to that observed in FIG. 1, the data in and data out buses **103**, **106** may be combined to form a bi-directional bus. Also, in various alternate embodiments, one of the enable lines **104**, **130** may be eliminated so that a single line is used to toggle the memory unit between being in a writable state and being in a readable state. Alternatively, it could be time based as in a shift register, or a mix of address and time based read. A commercial example that uses a mix of address and time based read/write is the Xilinx Virtex SRL primitive that is written into like a shift register and is read from like a RAM.

The memory unit **101** of FIG. 1 can be used to implement a number of storage related devices such as a random access memory (RAM), a first-in-first-out (FIFO) queue (e.g., by appropriately controlling the address values of the memory unit **101** such that a FIFO queue is emulated with the memory unit **101**), a content addressable memory (CAM), a shift register, etc. A problem with memory units (as they are offered to designers who wish to employ them in their circuit designs), however, is that they do not have a reset function. A reset function effectively "clears" the memory unit's cell word values to some "reset" value (e.g., a n wide value of "0"); and, often, the integration of circuitry for resetting the cell word values of the memory unit **101** is too expensive and/or complicated to implement. For example, according to one approach, in order to implement a resetable memory, each n wide storage cell is implemented with resetable flip-flops that are individually accessed via complicated multiplexing and control circuitry. Here, the use of resetable

2

flip-flops to implement each n wide storage cell (as well as the complicated multiplexing and control circuitry) can result in a resetable memory unit having noticeably slower performance (and that consumes more silicon surface area) than a memory unit that does not have resetable storage cells.

SUMMARY OF THE INVENTION

An apparatus is described that includes a memory without reset capability having a data output coupled to a first input of a first multiplexer. A second input of the first multiplexer has a reset value input. A channel select for the first multiplexer is coupled to a resetable storage cell output that indicates whether a storage cell within the memory without reset capability has been written to after a reset or has not been written to after a reset.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a memory unit.

FIG. 2a shows an embodiment of a resetable memory that is constructed with a memory unit that does not have a reset function.

FIG. 2b shows another embodiment of a resetable memory that is constructed with a memory unit that does not have a reset function.

FIG. 2c shows an embodiment of a register transfer level (RTL) description that may be used to describe the resetable memory of FIG. 2a.

FIG. 3 shows an embodiment of a methodology that can be executed by the resetable memory of FIG. 2.

FIG. 4 shows a design methodology that automatically incorporates a resetable memory that is constructed with a memory that does not have a reset function into a semiconductor chip design.

FIG. 5 shows another embodiment of a resetable memory that is constructed with a memory unit that does not have a reset function.

FIG. 6 shows another embodiment of a resetable memory that is constructed with a memory unit that does not have a reset function.

FIG. 7 shows an embodiment of a methodology that may be used to automatically install a resetable memory having a reset value write unit as observed in FIG. 6.

FIG. 8 shows an embodiment of a computing system.

DETAILED DESCRIPTION

A resetable memory is described that includes a memory without reset capability having a data output coupled to a first input of a first multiplexer. A second input of the first multiplexer has a reset value input. A channel select for the first multiplexer is coupled to a resetable storage cell output that indicates whether a storage cell within the memory without reset capability has been written to after a reset or has not been written to after a reset. Other methods and apparatus are also described.

Resetable Memory Unit Embodiment(s)

FIG. 2a shows an embodiment of a resetable memory **220** that is constructed with a memory unit **201** that does not have a reset function. The memory unit without reset **201** of FIG. 2a can be viewed as corresponding to the memory unit without reset **101** that was originally described in FIG. 1. The resetable memory **220** of FIG. 2a includes both a memory without reset **201** (as described above) as well as a

US 6,836,420 B1

3

memory unit **205** that includes a reset function (i.e., a memory unit **205** with reset).

In various embodiments, the data width of the memory unit with reset **205** is “smaller” than the data width of the memory unit without reset **201**. That is, in various approaches, the cell word size of the memory unit with reset **205** is less than the cell word size of the memory cell without reset **201**. In a further embodiment, the cell word size of the memory unit with reset **205** is one bit wide (i.e., such that its output line **210** carries only two states “1” or “0”).

As elaborated on in more detail below, the smaller memory with reset **205** cost effectively disguises the inability of the larger memory **201** to reset its cells. That is, the cooperative integration of the “smaller” memory with reset **205** and the “larger” memory with reset **201** allows the larger memory without reset **201** to appear (e.g., to downstream circuitry from data output **209**) “as if” it has a reset function; yet, the complexities/costs associated with integrating a reset function into the larger memory without reset **201** are avoided. As a result, an overall resettable memory **220** is constructed from the pair of memories **201**, **205** that has a reset function, yet is less expensive/complex than a memory unit of comparable storage capacity.

The resettable memory unit **220** may be viewed as having a memory unit data output **209** and an actual memory unit data output **206**. The actual memory unit data output **206** corresponds to the memory unit data output **106** described in FIG. 1. That is, for any read operation, the data word stored in the cell being addressed of the memory unit without reset **201** is observed at the actual memory unit data output **206**.

The memory unit data output **209** is the “effective” memory unit output as observed by the downstream circuitry that may use the data that is read from the resettable memory **220**. A multiplexer **207** is positioned between the memory unit data output **209** and the actual memory unit data output **206**. A first multiplexer **207** input receives the actual memory unit data output **206**. A second multiplexer **207** input receives a reset value **208**.

The multiplexer **207** is configured to present at the memory unit data output **209**: 1) the reset value **208** if the resettable memory unit **220** cell being addressed during a read operation has not been written to since its last “reset”; and 2) the data word read from the memory unit **201** (as provided on the actual memory unit data output **206**) if the resettable memory unit **220** cell being addressed during the read operation has been written to since its last “reset”. In this manner, the memory unit output **209** presents data consistent with the operation of a memory unit having the ability to reset the data words that it stores.

That is, after the resettable memory **220** has been “reset”, any attempt to read a data word from a particular cell within the memory unit without reset **201** will produce the reset value **208** at the memory unit data output **209**. This functional behavior continues for each cell until a particular cell is written to. After a particular cell is written to, the newly written data is presented at the memory unit data output **209** until the next reset of the resettable memory **220** occurs. The multiplexer **207** and memory unit with reset **205** may be viewed as providing for the emulation of a reset function for the memory unit without reset **201** reset.

In an embodiment where the data output **210** of the memory unit with reset **205** (which feeds the channel select input of multiplexer **207**) is one bit wide, the data output **210** effectively acts a state bit that signifies whether the resettable memory **220** cell being read from has been: 1) not written to since a last resettable memory **220** reset; or 2) has been

4

written to since a last resettable memory **220** reset. Note that in the particular embodiment of FIG. **2a** a “0” is used to signify the former and a “1” is used to signify the later.

The memory unit with reset **205** acts a storage resource that keeps a record, for each cell within the memory unit without reset **201**, as to whether or not the particular cell has been written to after a reset of the resettable memory **220**. The reset input of the memory unit with reset **205** acts as the reset input **219** for the resettable memory **220**. When the reset **219** is active, each cell within the memory unit with reset is cleared (e.g., is given a value of “0”). Noting that the Write Enable **204**, Address **202**, and Read Enable **230** lines of both memory units **201**, **205** are tied together, any attempt to read a particular cell within the memory unit without reset **201** causes a simultaneous read from the memory unit with reset **205** at its corresponding cell.

Thus, if an attempt is made to read a cell after a reset (and before any writing to the cell has occurred), the reset output value of the memory unit with reset **205** (e.g., a “0”) will be presented to the multiplexer’s **207** channel select input. According to the embodiment of FIG. **2**, a channel select value of “0” causes the reset value **208** to be presented at the resettable memory output **209**. As such, the resettable memory output **209** properly reflects a reset value for a cell that has not been written to after a reset has been applied.

Similarly, any attempt to write to a particular cell within the memory unit without reset **201** causes a simultaneous write into the memory unit with reset **205** at its corresponding cell. Noting that the data input **214** of the memory unit with reset **205** is tied to a “1” in the embodiment of FIG. **2**, the first writing to a particular cell within the memory unit without reset **201** will cause the corresponding cell within the memory unit with reset **205** to have its reset value (of “0”) to be written over (with a value of “1”).

Thus, subsequently, if an attempt is made to read the same cell before a next a reset occurs, an output value of “1” will be presented from output **210** of the memory unit with reset **205** to the multiplexer’s **207** channel select input. According to the embodiment of FIG. **2**, a channel select value of “1” causes the actual memory unit without reset output **206** to be presented at the resettable memory output **209**. As such, the resettable memory output **209** properly reflects the most recently written cell value for those cells that have been written to after a reset has been applied.

In this manner, the circuit of FIG. **2a** emulates the behavior of a memory unit having the storage capacity of memory unit **201** but also having reset capability. For each cell within the memory unit without reset **201**, once the resettable memory **220** is reset, the reset value **208** should be observed for all subsequent read operations until the cell is written to. After the cell is written to, the value of the most recent written data should be observed until the next reset of the resettable memory **220** occurs.

FIG. **3** shows a methodology that reviews the approaches discussed above. First, a reset is asserted **301**. This causes the reset value **208** to be provided **302** as the effective memory cell output—until the cell has been written to. If the cell has been written to, the actual value of the cell contents as stored within the memory unit without reset **201** are presented **303** as the output.

Before continuing note that the particular embodiment of FIG. **2a** indicates that the same reset value **208** is used for each resettable memory **220** cell. In alternate embodiments, different reset values may be used for various groupings of cells (or individual cells themselves). For example, referring to FIG. **2b** a reset value function circuit **230** may be inserted

US 6,836,420 B1

5

between (and coupled to) the reset value **208** input of the multiplexer **207** and the address input **202** of the resettable memory **220**. The circuit **230** may be designed to issue different reset values, depending on the particular addressing value that appears on the address input **202**.

Automatic Inference and Implementation Of Memory With Selective Reset

A further utility of the approaches discussed above is the ease at which a memory having reset may be incorporated into a designer's circuit design—even if the underlying semiconductor technology does not easily provide for a memory core having selective reset. For example, semiconductor circuits are typically designed with a particular semiconductor manufacturing process (i.e., a “foundry”) in mind. Usually, the foundry supplies models of basic building blocks (e.g., logic gates, memory units, etc.) from which a semiconductor chip design can be constructed.

The resettable memory approach discussed above may be used to effectively provide a designer with a resettable memory, even though the foundry only manufactures traditional memory units without reset (such as the traditional memory unit discussed with respect to FIG. 1); or, offers resettable memories that are too cumbersome to implement as a large memory. For example, in one embodiment, a software design tool may be configured to offer the insertion of a resettable memory (such as that described above in FIG. 2) into a designer's design. If the designer decides to incorporate a resettable memory into his or her design, the resettable memory is automatically “built into” the designers design.

In one embodiment, the use of a resettable memory is automatically inferred from the behavioral level or RTL level description of the designer's design; and, in response, a resettable memory is automatically inserted into the circuit designer's design. A behavioral level or RTL level description is a circuit description that is tailored to be understood by a computer and that describes the circuit in terms of its methodology (e.g., the various processes performed by the circuit and the relationship(s) between them) as opposed to describing the circuit only in terms of its hardware components and the interconnections between them (e.g., gates, registers, signal lines, etc.).

Given that behavioral or RTL level descriptions are written in terms of methodologies to be understood by a computer, they are frequently documented in a form that is similar to a software program or a pseudo code description of the operational flow of the circuit. As an example, FIG. 2c provides an embodiment of an RTL level description that can be used to describe the resettable memory of FIG. 2a. In order to automatically “install” a resettable memory into a circuit designer's design, a software design tool can be configured to automatically infer the use a resettable memory from a behavioral level or RTL level description by identifying from the operational flow of the circuit description that a reset condition is being individually applied to one or more variables (e.g., that is to be implemented in hardware as a data word that resides within some type of storage cell).

Furthermore, in response to the inferred use of a resettable memory, the software tool may be further designed to automatically install (or present the designer with an offer to automatically install) a resettable memory into the designer's circuit. In one embodiment, the resettable memory **220** is incorporated into the designer's circuit at the gate level. Typically, an RTL level description of a circuit is compiled into a gate level netlist that describes the specific digital

6

structures used in the design and the interconnections between them. Thus, upon recognition of a reset condition within the behavioral level or RTL level description of the circuit, the compilation result into its corresponding gate level form is modified by the design tool so as to incorporate the resettable memory.

In a further embodiment, if a memory unit with reset is offered by the applicable foundry, the foundry offering is incorporated into the resettable memory as the memory unit with reset **205**. Further still, the design tool may be configured with its own design library that includes a memory with reset **205** that can be configured from a foundry's standard logic offerings. As result, the resettable memory **220** can be constructed even for those foundries that do not offer a memory with reset in their standard offerings. An embodiment of a gate level memory with reset that can be constructed from standard logic is provided in more detail ahead.

FIG. 4 shows a methodology that reviews such a design flow sequence. First, the use of a resettable memory is automatically inferred **401** from the designer's behavioral or RTL level description. The automatic inference can be accomplished, for example, by configuring the design tool to recognize from the operational flow of the circuit that: 1) some type of reset is being applied to the stored data values within the circuit; and 2) the stored data values are being changed to some type of reset value in response. After the use of a resettable memory has been inferred **401**, a resettable memory is incorporated into the designer's design (e.g., at the gate level as described just above).

Note therefore that embodiments of the present approach (as well as those discussed further ahead) may be implemented not only within a semiconductor chip but also within machine-readable media. For example, the designs discussed above may be stored upon and/or embedded within machine readable media associated with a design tool used for designing semiconductor devices. Thus, it is also to be understood that embodiments of this invention may be used as or to support a software program executed upon some form of processing core (such as the CPU of a computer) or otherwise implemented or realized upon or within a machine-readable medium. A machine-readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine readable medium includes read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

Resettable Memory With Gate Level Design For Memory Unit With Reset

FIG. 5 relates to resettable memory design embodiment that can use standard foundry logic to create a memory unit with reset. As such, the design of FIG. 5 can be used even if the underlying foundry does not provide a memory with reset in its standard design library. According to the design embodiment of FIG. 5, a resettable register **511** is used as a storage cell for the memory unit with reset. Here, the register **511** holds the value of a state bit that controls the channel select of the multiplexer **507** for its corresponding cell.

The register **511** also has a reset input **519** used to effectively reset the storage cell. When a reset signal is presented to the register **511**, the state bit from the register output is set to a “0”. As such, the channel select line **510** of

US 6,836,420 B1

7

the multiplexer 507 is positioned to enable the reset value input 508 of the multiplexer 507. According to the design approach of FIG. 5, the register 511 output is coupled in a feedback arrangement to an input of a second multiplexer 513. The other input 514 of the second multiplexer 513 corresponds to a logical value that is opposite that of the register 511 reset value. That is, because the register 511 reset value is "0" in the embodiment of FIG. 2, a "1" is provided to the input 514 of the second multiplexer 513.

Note that, after a reset, the feedback input to the second multiplexer 513 will be a "0" because the register 511 reset value is "0". The second multiplexer 513 is configured to transition from the enablement of the feedback input to the enablement of the "1" input 514 when the memory unit cell is first written to after a reset. That is, for as long as the memory unit cell is not written to after a reset (i.e., is either reset again, read from or not used at all), the write enable (WE) line 504 and ADDR_HIT line 517 remain at a logic "0".

As such, during this time period, the value of the channel select line 515 for the second multiplexer 513 is a logic "0". This corresponds to the register 511 being continuously latched with its reset output value of "0". As described above, this further corresponds to the channel select line 510 of the first multiplexer 507 being positioned to enable the reset value input 508 of the multiplexer 507. In the embodiment of FIG. 5, the WE line 504 is a logic "1" only if a write operation occurs; and, the ADDR_HIT line 517 is a logic "1" only if the address corresponding to the memory unit without reset 501 cell represented by register 511 appears at the address input 502.

As such, the output 515 of the AND gate 516 is a logic "1" only if the memory unit without reset 501 cell represented by register 511 is being written to. When this occurs, the "1" input 514 to the second multiplexer 513 is enabled, and a "1" is latched into the register 511. Thus, due to the feedback input of the second multiplexer, a "1" is continuously latched into the register 511 after the first write operation is performed upon the memory unit cell (after it has been a reset); and, a "0" does not appear at the register 511 output until it is next reset.

This corresponds to the actual memory unit data output 506 being enabled by the first multiplexer 507 after the memory unit cell is first written to (after it has been reset). The actual memory unit data output 506 is then enabled until the memory unit cell is reset again. Before continuing, note that the ADDR_HIT 517 may be provided by a logic circuit (not shown in FIG. 5 for simplicity) that effectively decodes the address input 502 value.

For simplicity only one register 511 is shown in FIG. 5. However, it should be understood that each of the various inputs of multiplexer 550 may be coupled to its own register and corresponding circuit. That is, for example, the circuitry outlined by region 560 may be repeated at each multiplexer 550 input to provide reset capability for a plurality of cells. Note that the ADDR_HIT input 517 associated with each repeated circuit should register a logical "1" only when the particular memory unit cell that the circuit applies to is accessed.

The third multiplexer 550 collects each register output. During a memory read from a memory unit cell having selective reset capability, the address of the memory unit cell to be read is used as an input to the third multiplexer 550 so that the reset state bit from the appropriate register (i.e., the register that corresponds to the memory unit cell being read from) is delivered to the first multiplexer 507. As such, the

8

channel select 510b of the first multiplexer is controlled on a "per cell" basis.

First Alternate Embodiment of Resetable Memory

FIG. 6 shows another resetable memory embodiment 620. The resetable memory 620 of FIG. 6 includes a memory unit without reset 601 and a reset value write unit 640 and reset input 619 that are coupled to the write enable 604, address 602 and data input 603 of the memory unit without reset 601. According to one approach, in response to an active reset being applied at the reset input 619, the reset value write unit 640 writes a reset value into each cell of the memory unit without reset 601. Thus, similar to the approaches described above, the memory unit without reset 601 "appears" to circuitry downstream from output 606 "as if" it is resetable.

Because an extended amount of time may be consumed by the reset value write unit 640 in the writing of a reset value into a number of memory unit without reset 601 cells, the resetable memory embodiment 620 of FIG. 6 is most appropriate in those environments where a write operation to the memory unit without reset 601 does not occur for an extended period of time after the reset input 619 is activated. As such the write enable input 604, the address input 602 and the data input 603 are sufficiently free to accept the input signals provided by the reset value write unit 640 (while reset values are being written into the memory unit without reset 601) without interruption from sources outside the resetable memory 620. The reset value write unit 640 may be designed, as just one embodiment, to repeatedly: 1) increment a prior address value so as to form a current address value; and, 2) write a reset value at the current address value.

Here, an appropriate environment for the type of resetable memory 620 observed in FIG. 6 can be determined from an analysis of the number of available clock cycles that may be used to execute a reset. As clock cycles can be inferred from within a behavioral or RTL level environment, the above described analysis can be performed at the behavioral level or RTL level as well. Generally, the more clock cycles that may be consumed in executing a reset, the more likely it is that the approach of FIG. 6 can be successfully implemented.

As such, a design tool may be configured to not only infer the use of a resetable memory from a behavioral or RTL level description, but may also be configured to infer the number available of clock cycles for executing a reset to one or more variables. This may be accomplished, for example, by estimating or calculating the time (e.g., by counting the number of clock cycles) between the moment a reset is issued to a particular variable; and, the earliest moment when the reset variable could be subsequently needed by the operational flow of the circuit

Furthermore, a determination as to the amount of time that is available to implement a reset may be used to drive which particular type of resetable memory (e.g., a resetable memory 220 having a memory unit with reset 205 as observed in FIG. 2a; or, a resetable memory 620 having a reset value write unit 640 as observed in FIG. 6) is to be automatically installed by a software design tool. FIG. 7 shows an embodiment of a methodology that describes such an approach in more detail. Here, as in the methodology of FIG. 4, the existence of a resetable memory is inferred 701 from the operational flow of a behavioral or RTL level circuit description.

Then, the number of available of clock cycles available for implementing a reset are inferred 702. Here, if the

US 6,836,420 B1

9

number of clock cycles is small enough so as to be less than a threshold value "X" (which, in turn, corresponds to the amount of time consumed by the reset value write unit 640) then the use of a resettable memory having a reset value write unit 640 is de-emphasized (because the reset value write unit 640 may need more time to execute the reset than the circuit can wait to use the reset value). As such, a resettable memory having a memory unit with reset and a memory unit without (e.g., as observed in FIG. 2a) is automatically installed 705. By contrast, if the number of clock cycles is large enough so as to be greater than the threshold value "X" then the use of a resettable memory having a reset value write unit 640 is emphasized (because the reset value write unit 640 has a sufficient amount of time to execute a reset). As such, a resettable memory having a reset value write unit 640 (e.g., as observed in FIG. 6) is automatically installed 704. Here, often, the approach of FIG. 6 consumes less semiconductor chip surface area than the approach of FIG. 2a; and, as such, in an effort to efficiently consume semiconductor surface area, the methodology of FIG. 7 installs the approach of FIG. 2a only if the approach of FIG. 6 is deemed to be unfeasible.

FIG. 8 shows an embodiment of a computing system 800 that can execute instructions residing on a machine readable medium (noting that other (e.g., more elaborate) computing system embodiments are possible). The instructions may be related to integrated circuit design (e.g., as described in FIGS. 2a through 7). In one embodiment, the machine readable medium may be a fixed medium such as a hard disk drive 802. In other embodiments, the machine readable medium may be movable such as a CD ROM 803, a compact disc, a magnetic tape, etc. The instructions (or portions thereof) that are stored on the machine readable medium are loaded into memory (e.g., a Random Access Memory (RAM)) 805; and, the processing core 806 then executes the instructions. The instructions may also be received through a network interface 807 prior to their being loaded into memory 805.

In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

What is claimed is:

1. A method, comprising:

a) inferring the existence of a resettable memory from a behavioral or RTL level description of a semiconductor circuit; and

b) incorporating a resettable memory design into a design for said semiconductor circuit.

2. The method of claim 1 wherein said incorporating a resettable memory design into a design for said semiconductor circuit further comprises incorporating a gate level resettable memory description into a gate level description of said semiconductor circuit.

3. The method of claim 1 wherein said resettable memory design comprises a memory without reset capability having a data output coupled to a first input of a first multiplexer, a second input of said first multiplexer having a reset value input, a channel select for said first multiplexer coupled to a resettable storage cell output that indicates whether a storage cell within said memory without reset capability has been written to after a reset or has not been written to after a reset.

10

4. The method of claim 3 wherein said storage cell is a storage cell within a memory unit having reset capability.

5. The method of claim 3 wherein said storage cell is a register.

6. The method of claim 1 wherein said resettable memory design further comprises a memory unit without reset and a reset value write unit that writes a reset value into storage cells of the memory unit without reset after a reset is applied.

7. The method of claim 1 further comprising inferring a number of clock cycles available for a reset between said inferring the existence and said incorporating.

8. The method of claim 7 wherein said incorporating a resettable memory design further comprises incorporating a resettable memory design having a first memory unit without reset capability and a second memory unit with reset capability if said number of clock cycles corresponds to a first amount of time that is less than a second amount of time in which a reset value can be written into a memory unit without reset capability.

9. The method of claim 7 wherein said incorporating a resettable memory design further comprises a resettable memory design having a reset value write circuit coupled to a memory unit without reset capability if said number of clock cycles corresponds to a first amount of time that is greater than a second amount of time in which a reset value can be written into said memory unit without reset capability.

10. The method of claim 1 wherein said inferring the existence of a resettable memory further comprises identifying within an operational flow of said description that a reset is being applied to a variable.

11. A machine readable medium having stored thereon a sequence of instructions which, when executed by a digital processing system, cause said system to perform a method, said method, comprising:

a) inferring the existence of a resettable memory from a behavioral or RTL level description of a semiconductor circuit; and

b) incorporating a resettable memory design into a design for said semiconductor circuit.

12. The machine readable medium of claim 11 wherein said incorporating a resettable memory design into a design for said semiconductor circuit further comprises incorporating a gate level resettable memory description into a gate level description of said semiconductor circuit.

13. The machine readable medium of claim 11 wherein said resettable memory design comprises a memory without reset capability having a data output coupled to a first input of a first multiplexer, a second input of said first multiplexer having a reset value input, a channel select for said first multiplexer coupled to a resettable storage cell output that indicates whether a storage cell within said memory without reset capability has been written to after a reset or has not been written to after a reset.

14. The machine readable medium of claim 13 wherein said storage cell is a storage cell within a memory unit having reset capability.

15. The machine readable of claim 13 wherein said storage cell is a register.

16. The machine readable medium of claim 11 wherein said resettable memory design further comprises a memory unit without reset and a reset value write unit that writes a reset value into storage cells of said memory unit without reset after a reset is applied.

17. The method of claim 11 further comprising inferring a number of clock cycles available for a reset between said inferring the existence and said incorporating.

US 6,836,420 B1

11

18. The method of claim 17 wherein said incorporating a
resetable memory design further comprises incorporating a
resetable memory design having a first memory unit without
reset capability and a second memory unit with reset capa-
bility if said number of clock cycles corresponds to a first 5
amount of time that is less than a second amount of time in
which a reset value can be written into a memory unit
without reset capability.

19. The method of claim 17 wherein said incorporating a
resetable memory design further comprises a resetable 10
memory design having a reset value write circuit coupled to

12

a memory unit without reset capability if said number of
clock cycles corresponds to a first amount of time that is
greater than a second amount of time in which a reset value
can be written into said memory unit without reset capabil-
ity.

20. The method of claim 11 wherein said inferring the
existence of a resetable memory further comprises identi-
fying within an operational flow of said description that a
reset is being applied to a variable.

* * * * *

CERTIFICATE OF SERVICE

I hereby certify that I electronically filed the foregoing with the Clerk of the Court for the United States Court of Appeals for the Federal Circuit by using the appellate CM/ECF system on December 23, 2015.

I certify that all participants in the case are registered CM/ECF users and that service will be accomplished by the appellate CM/ECF system.

ORRICK, HERRINGTON & SUTCLIFFE LLP

/s/ Eric A. Shumsky

Eric A. Shumsky

Counsel for Appellant

CERTIFICATE OF COMPLIANCE

This brief complies with the type-volume limitation of Fed. R. App. P. 32(a)(7)(B)(i) because this brief contains 10,394 words, excluding the parts of the brief exempted by Fed. R. App. P. 32(a)(7)(B)(iii).

This brief complies with the typeface requirements of Fed. R. App. P. 32(a)(5) and the type style requirements of Fed. R. App. P. 32(a)(6) because this brief has been prepared in a proportionally spaced typeface using Microsoft Word 2010 in Century Schoolbook 14-point font.

ORRICK, HERRINGTON & SUTCLIFFE LLP

/s/ Eric A. Shumsky

Eric A. Shumsky

Counsel for Appellant